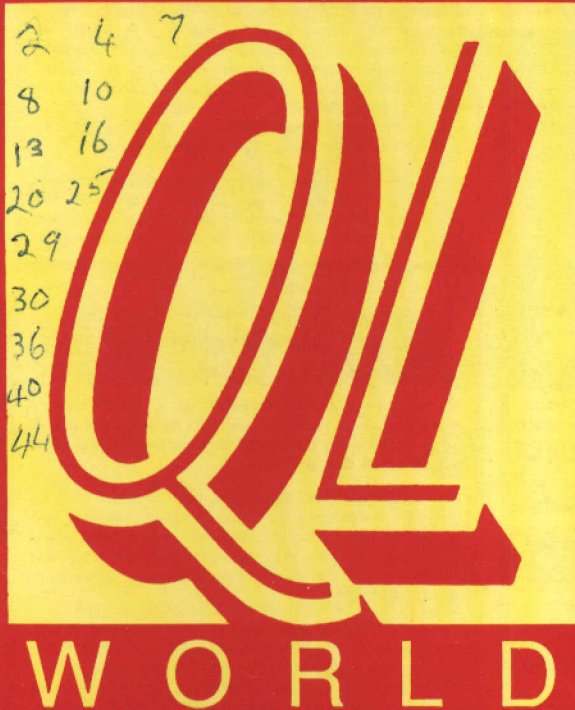


SINCLAIR

APRIL 1992 £1.95



## THREE IN THE HAND

Meeting the three best QL word-processors

## MORE THAN JUST A MIRACLE

Miracle Systems is going for the big one

HALF A LAPTOP  
Importable, exportable  
... or just portable!





SINCLAIR



*Editor*

Helen Armstrong

*Production Controller*

Jayne Penfold

*Advertising Manager*

Jason Newman

*Magazine Services*

Yvonne Taylor

*Advertising Production*

Michelle Evans

*Group Advertising Manager*

Lynda Elliott

*Group Editor*

John Taylor

*Publishing Director*

Wendy Palmer

*Deputy Managing Director*

Ray Lewis

*Managing Director*

Peter Welham

**Sinclair QL World**

**Panini House**

**116-120 Goswell Road**

**London EC1V 7QD**

**Telephone 071-490 7161**

**ISSN 026806X**

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write the The Editor, Open Channel, Trouble Shooter, or Pision Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.

Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request.

Published by Headway, Home and Law Publishing Ltd., Sinclair QL World is distributed by IPC Market force, King's Reach Tower, Stamford Street, London SE1 9LS. Subscription information from: HHL Subscription Department, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN. Tel: 091 510 2290

UK: £21.00, Europe: £32.00, Rest of the World: £38.00.

Please include your subscriber number with any queries.

Typesetting by Ford Graphics, 8-10 Whitsbury Road, Fordingbridge, Hampshire. SP6 1BR. Tel: (0425) 655657 Printed and bound by BPCC, Colchester. Covers printed by Spottiswoode Ballantyne, Colchester. Sinclair QL World is published on the third Thursday preceding cover date.

© COPYRIGHT  
SINCLAIR QL WORLD - 1991

# CONTENTS

■ ■ APRIL 1992

- 4 TROUBLESHOOTER** ● Bryan Davies focusses on letters this month
- 11 SOFTWARE FILE** ● QL Genealogist
- 12 QL SCENE** ● Bio-powered QL?
- 14 OPEN CHANNEL** ● On the move
- 15 QL SCENE** ● Euro distribution for Digitiser
- 16 HITTING THE GOLD** ● Meet Miracle Systems
- 19 SOFTWARE FILE** ● Picture Master Plus
- 22 HALF A LAPTOP** ● A new use for the Z88
- 25 THE NEW USER GUIDE** ● DIM TO ED #chan
- 29 SOFTWARE FILE** ● Filemaster
- 30 THREE IN THE HAND** ● Three famous wordprocessors
- 36 DIY TOOLKIT** ● Four functions and two ideas
- 40 SOFTWARE FILE** ● DEA and DISA assemblers
- 43 SOFTWARE FILE** ● Remind Me
- 44 SYSTEMATIC MACHINE CODE PROGRAMMING** ● Part 6



## NEXT MONTH

We have reviews of *3D Terrain*, *Polytext*, *Quickdisk* and *MasterBasic* lined up, and also the long-awaited review of the *GOLD CARD* by Mike Lloyd. Also another *Archive Answers*, and more on desktop publishing.



# T A R O U B L E

**A**s it's the April issue, a suitably over-the-top suggestion seems appropriate. What about buying a colour copier? Too expensive, yes, but the rate at which prices are coming down suggests that we may be able to contemplate such exotica before too long. The one that caught my eye was from Brother, and costs £1,995 plus VAT. A year or so ago, you wouldn't have seen the like of it for five times the price. The same applies to colour printers, which are rapidly coming down into 'our' range. Despite the way goods are advertised, buyers' minds don't usually work only on percentages; if something is reduced by 50%, from £10,000 to £5,000, you won't be rushing out to buy it, but what if a laser printer comes down 50% from £500 to £250? That may have you reaching for your wallet. Laser printers – the black-and-white type – have all but reached that price level now, and colour is not that far behind.

## Spending money

The add-ons are what you spend your money on, not the computer. Whether you paid the original £399, or the subsequent £199 or less for your QL, the chances are that you will have spent more than that on the bits and pieces you've added to it since. Provided the software suppliers keep up with the times, you will still be able to retain the same 'it doesn't owe me a penny' QL to produce the output for whatever printer you buy in future. There's a good chance you'll need a lot of memory and plenty of patience to get prints that look as good as the cover of this magazine, but the memory can at least be purchased.

The WDEL command is one of my pet irritations at the moment. If memory serves aright, the problems with this command are fairly recent phenomenon, and it is necessary to consider the possibility that hard disk, Gold Card, ED drives, and later rom versions of the extra commands, are involved in what is now happening. The command *appears* to work; it puts up the usual Y/N/Q/A? type of message, and goes away without murmur after the answer is given. The trouble is that a subsequent DIR reveals that some or all of the 'deleted' files are still on the disk. The erratic Last Line Recall function might

## Bryan Davies goes briefly over the top – and then settles down in his Troubleshooting hat.

have been put down to a peculiarity of the JM rom, or the Trump Card, except that it works every bit as badly with a JS rom and Gold Card. Wouldn't it be nice if someone cleaned up the whole command set, and put it all in one *disk* package that doesn't require 50 KB to load and gives a consistent QL?

Yet another removal of the hard disk allowed the Ice rom to be plugged into my system again. The 'desktop' interface may be lacking in some features, but it is still very useful, and straightforward to use. Years ago, I went to much trouble to create icon macros with the Iclicle program, to simplify preparation of my accounts, and they served me well until the enforced removal of Ice made them unavailable. It's good to have them back but, like so many of our better tools, Ice is not properly compatible with other things that have a good claim to be fitted.

## Readers' letters

**Noel Mortier** reports having received from **TK Computerware** the cartridges that were the subject of correspondence some months ago. The difficulty **H.F. Banks** had with running two pairs of drives, with different densities, off Trump Card and Gold Card appears to have been partially solved, and he should by now have upgraded his system to ED drives from **Miracle Systems**. He may not be alone in deciding that the fitting of upgrade rom chips is something best left to experts, especially when one is getting on in years. From correspondence, it would seem that quite a few QL users are drawing their pensions; a lot more are likely to be in the age group at which you suddenly realise that a pension is looming close!

Back on the Serial 8056 again, **PJ Hud-**

**son** is the latest user to suggest a way around the problem of obtaining paper for this printer. He has been using fax paper for three years and says it works fine in the 8056. The supplier he suggests (see **INFORMATION** below) can provide 30, 50 and 100 metre lengths in both 210 and 216 mm widths, at prices ranging from £2.93 for a 30 m roll of 210 or 216 mm width, to £42.23 for 10 rolls of 100 m length at 210 mm width. These prices include VAT but not post and packing. Presumably, the paper does not come on the same size roll that the 8056 uses, as Mr Hudson retains his old rolls and re-winds new paper onto them.

## Roland Piano

**Norman Foster** enquired if we have any expertise on the subject of connecting a Roland electronic piano to the QL, via a MIDI interface. My own knowledge in this area is zero – has anyone done this? His other request was for comment on the advisability of replacing a Trump Card with a Gold Card. Money has to be the major factor for most of us, but there is a decent trade-in allowance for users who currently have a Trump Card. My experience with a Gold Card leaves me in no doubt that it is a desirable upgrade; of its various advantages, speed is the one that strikes you first, and most strongly. Programs which were never noted for their speed become quite nippy; I've just been running the *Conqueror* emulator and using the PC wordprocessor *WordPerfect* 5.0 and have been agreeably surprised not to have to wait for my typed words to appear on the screen. More to the point for most users, *Quill* handles even large documents in an acceptably fast manner, the cursor moving at a sensible speed and the Copy command working well (unless you ask it to make a copy 50-100 pages away!). The 2 MB of ram will cover most users' needs until retirement.

## Archive loss

In the January issue, **J Paul Bissonette** described his loss of *Archive* files, apparently through a bad floppy disk. You can't diagnose faults such as this at long range,



# SHOOTER

M S O L V E D

but it strikes me that a faulty disk is not the place to look for such troubles. There are reasons for disks themselves going bad – rough handling and consequent mechanical damage (including the pouring of coffee into them!), exposure to excessive heat, being placed in close proximity to strong magnetic fields, physical impact of the read/write head on the disk surface, etc. The user would normally be aware of such things, though, and would experience trouble with other disks that had been treated the same way.

The fact that one 'bad' disk didn't appear to have any bad sectors when formatted on another computer and drive, looks at first sight to absolve it from blame, but it was not clear from the letter whether or not the same disk subsequently showed bad sectors when checked in the same drive after not being used at all for some days, or whether it behaved badly when used again in the same QL drive. My own experience with close to a thousand floppy disks of both various sizes and formats, over more than five years, is that errors are exceedingly rare when disks are treated reasonably, and are quite obvious from the first time a disk is used; for example, a bad disk may fail to be formatted to the full number of sectors straight off (it may subsequently format correctly, but not consistently). If a commercially supplied program disk is bad, errors are fairly sure to show up as soon as programs are run.

## Format fails

When a format operation on a floppy fails once, the disk may still be usable. Failure of the format operation more than once should be taken as a sign that the disk is not safe for valuable files, assuming that you are sure there is no problem with the drive. If more than one disk cannot be formatted correctly in the same drive, the chances are that the problem lies with the drive. With hard disk drives, the situation is essentially the same, although there are differences arising from the difference in working conditions. Physical or magnetic damage to a hard disk is unlikely, because of its sealed construction and secure location within a computer. You can, however, do damage to the magnetic platters and/or read/write heads inside a hard drive by treating the computer to a large shock,

such as being dropped on the floor. You are fairly certain to find out as soon as you use the computer and drive afterwards, though. Bad sectors are not at all uncommon on hard disks – from the specimens in my collection, trouble from bad sectors is **very** much more common with hard disk than with floppies, and it is essential to have a utility program which checks for, and marks, bad disk areas. The QL is poorly served in this respect; it is not easy to identify, or mark, bad areas on floppies or hard disks. If Mr Bissonette is still having disk problems after all these months, maybe he would care to send a sample disk for examination here?

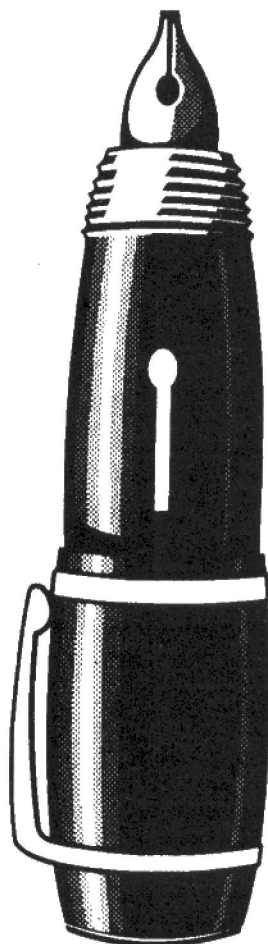
tion instruction manual. As a general point, it should be said that even DP's extensive manuals are diminutive compared to PC manuals, and the user should be grateful for that. Better a smaller manual that one actually reads all the way through, than an exhaustive one that never, ever gets read fully. Having said that, I must add that there were various items in the *Perfection* manual which I found difficult to find at first, but some changes have been made to the manual since that time. A much-needed index has been added.

## Reformatting

On the subject of reformatting text, it seems we are in a very subjective area, and my own experience is that the Quill procedure of automatically reformatting text 'as it goes' is the norm, rather than the exception. We have to talk about other computers as well as the QL here, otherwise there are only three programs to compare – *Quill*, *text*<sup>67</sup> and *Perfection*. Quill performs normal, full reformatting actions, when you insert additional text into existing text, and when you alter margin settings. It is strange, in that it will allow you to go back and change left and right margins for a single paragraph but any change of indent margin affects subsequent paragraphs also. *Text*<sup>67</sup> reformats fully when insertions are made, but is less certain about what to do with changes of margin; changes take full effect from the point at which they are made, and it implements left margin changes fully in existing text also, but it doesn't always do the same with right margin changes. The Reformat command has to be used to make sure changes have been implemented completely.

## Perfection

Perfection also leaves you rather unsure of what has changed, as you can insert some text and have it reformatted, but not always completely; with changes of margins, they are fully implemented from the point at which they are made – for new text – but have no effect on text before the cursor (this was also normal PC behaviour) or on existing text if the cursor has been moved back before the changes are



In that same issue of *QL World*, there was some comment about wordprocessing programs, in connection with **Capt. Starling's** earlier comments about the *Perfec-*



made. Again, you need to use the Reformat command to make sure changes take effect everywhere. What puzzles me is the *reason* for making reformatting a non-automatic operation. This has never seemed to be something that is ever required of a WP program, even as an option, let alone as standard procedure. Presumably it comes from the world of programming.

### QL in Italy

As mentioned in a previous issue, QItaly is an Italian QL user group, and two disks containing issues of their regular magazine were sent to me some while back. Looking at the contents of these disks was by no means a five-minute job and comment has had to wait until now. The magazine is very well prepared, professionally presented, and contains a lot of information. One issue was devoted to *QRam/QPac* and related software, covering the work of Tony Tebby and Jonathan Oakley at QJump, and the later developments from (mainly German) software writers/suppliers such as Jochen Merz, and the brothers van der Auwera. The opening screens of these disks are illustrated here. There are a host of screen

files, showing aspects of the Pointer Environment software, and plenty of text files giving information on it. There is no need to fiddle about to see what is of interest – the boot file takes even an inexperienced



user by the hand, through all that is on the disk. Obviously, plenty of thought has gone into the little details that can mar such an effort if not attended to.

It requires considerable enthusiasm to sustain any magazine for more than a few

months. The presentation of the QItaly magazine is very good. What is surprising is that this particular magazine has survived 19 (maybe more, by now) issues, over a period of 3 years, and is still going strong so far as I know. Congratulations to QItaly!

### New text<sup>87</sup>

As of early February, the copy of version 4 of *text*<sup>87</sup> promised several months ago has not materialised and one has to assume that some of the upgraded features in the program are taking much longer to implement than expected. In addition, the programmer himself had a bug for some weeks after Christmas! While nobody is happy with delays, they are by no means unusual with software. Trying to match added features in competitive programs, and cater for new hardware (eg GC and ED drives, hard disk), must be a wearing job.

### INFORMATION

#### Serial 8056 FAX paper:

"Office World", address unknown, but branches in Nottingham, Leicester and Northampton.

## Jochen Merz Software

Im stillen Winkel 12 - W-4100 Duisburg 11 - Germany

☎ & Fax: 0203 501274

**QD III** - The first (and only) Editor using the Pointer Environment. £38

**NEW!!! QMenu** - the Menu Extension. NOW IN VERSION 3! QMenu is a very easy to use interface with pre-defined menus (e.g. file-select, simple-choice boxes, select from lists). These menus may be used from SuperBASIC, machine code and other languages. File-select, directory-select and list-select all re-written with more options, multi-columns etc. **£12.90 Update from previous version with new manual £5**

**FIFI** - the FileFinder FiFi is easy to use and extremely useful. FiFi scans devices or directory trees and searches files or filenames for strings. Combinations are possible, e.g. String 1 AND String 2 BUT NOT String 3 etc. **£16**

**DISA** - Intelligent Disassembler which allows you to disassemble interactively any kind of machine-code. Generates ASCII-output which may be assembled using the GST (Macro) Assembler. **£29**

**QDOS Reference Manual** - This book is a must for all m-code programmers. It explains how to use QDOS, all traps and vectors, the Thing System, the HOTKEY System II and much more. It points out which features work on a QL, an Emulator, how to write compatible for future operating systems. 170 pages. **£30**

**QPTR** - The Pointer Environment Toolkit Revised manual which describes how to use the Pointer Interface and the Window Manager from SuperBASIC and machine-code. QPTR comes with examples on disc, keys, macros and extensions for SuperBASIC. **£30**

**FLP/RAM Level 2 now with ATRI!!!** Replacement EPROM for SuperQBoards (V1.17 onwards, with & without mouse) or TrumpCard (please specify!). New Manual! Real sub-directories! QL-Emulator compatible! About twice as fast, with improved slave-block-handling! **£18**

**EASYPTR II** - Create your own Pointer-Env. menus & sprites, use them in your own SuperBASIC or machine-code programs. Supports all the PE facilities, even split application sub-windows and uses the Menu Extension! Many examples and better manual! **£49**

**NEW!!! SYSTEM** - System Tools for QDOS: real file-attributes are added, e.g. write-only, hidden or for selected users only. Works even over net and on winchester. New TRAP#3. New devices (equiv. to QL-Emu.): named pipes, MEM, NUL 80 new procs & features, e.g. WSET\_FATTR, WSET\_FTYPE, WSET\_FVER5, ADIR, ASTAT, HK\_GETBFS **£27.50**

### QL-Emulator for ATARI ST

(260, 520, 1040 and all Mega's, but no STE's). This hardware, together with software turns any ATARI into a QL. The emulator supports the standard MODE 4 as well as a MODE 4 in much higher resolution (768x280)

It supports real subdirectories (complete Level2 device drivers) on floppy, hard-disk and RAM-disc.

The complete QL emulator package includes a lot of utility software and extensions. If you are interested in an emulator, you should write for more information **£166**

**EPROM Cartridge for the QL-Emulator, switchable**

- with standard drivers **£33** - with extended drivers **£33**

**Floppy adaptor to connect QL discdrives to the ST £9**

**Monitor-cable ATARI to Scart £9**

**Centronics-cable 2m £6.50 5m £10.50**

### Games

Diamonds £11 - BrainSmasher £12 - Arcanoid £10

Firebirds £10 - SuperGamesPack £25 - QShang £14

**NEW!!! The Lonely Joker: 3 Patience games**, many options, within one program (maybe you know it from Windows). It runs under the Pointer-Env. Very nice card symbols, very addictive, keeps you busy for hours. **£14.90**

**Pack of 10 3.5" discs MF2DD £3.10 - MF2HD £5**

**QSUP System Utility Pack £26 - Thing & Eprom Manager £18.50**

**QDesign** - Graphic and Design-program (9 & 24 pin) **£38** (Deskjet & Laserjet) **£45**

**Font-Vector-Editor for QDesign £19**

**DataDesign** - Database program **£55**

**Please add £4.00 for postage and package (Europe) or £4 for one item and £2.50 for every further item (Overseas).** All prices excl. V.A.T. E&OE.



Cards welcome! No extra charge!



# SOFTWARE FILE

## John Shaw calls on his mighty ancestors.

**W**hen I was reviewing *QL Genealogist* in the January 1991 edition of *QL World* I found myself almost running out of superlatives in my efforts to describe the program. Chris Boutal seemed to have created an almost perfect family research program. I could think of no other way in which it could be improved. Now, to my delight, a Second Edition has emerged, enhancing one of the jewels in the QL's crown.

I think that if I had created such a program, I might have rested on my laurels, but not so Chris. What has he done to improve the original *QL Genealogist*? The answer virtually fills 22 pages of an A4 updates manual. I can't detail everything, but I will describe some of the more important changes.

One of the most important features of any complex program is the instruction manual. If this is difficult to understand, the whole thing becomes difficult to use. Chris has taken great pains to make sure his manual is easy to follow, logical and helpful. Even a total newcomer to the QL or family history research will have little difficulty in becoming proficient. A step by step tutorial covers four pages and provides all that is needed to use every aspect of the program.

### Data safe

Owners of the original *QL Genealogist* will be pleased to hear that their data needs no modification to work on the Second Edition. Chris, however, warns that once *SAVED* by the Second Edition, the files cannot be read again by the original version.

### INFORMATION

**Program:** *QL Genealogist*:

*Second Edition*

**Supplier:** Dilwyn Jones

Computing,

41 Bros Emrys

Tal-y-Bont

Bangor, Gwynedd

**Price:** *QL Genealogist*:

*Second Edition*

£30.00 disk only;

Budget 128K *QL*.

*Genealogist* £10.00

disk £12.00 mdv;

Upgrade Standard

to *Second Edition*

£12.00 disk only;

Upgrade Budget

to *Second Edition*

£24.00 disk only;

Translation service

from *Archive* £6.00.

Minimum 256K memory expansion required (except for 128K Budget edition).

Runs on Minerva and

Goldcard

In the main menu, Spelling is now moved from Research so that it can be used in the Family Tree section. A Colours command allows you to change screen colours according to your whim. Two new Export options are provided; All Tree data can now be exported to *Abacus*.

Geography (new feature) data can be exported to *Easel*.

In the Family Tree section, the author has made use of the five function keys to greater effect than before. <F1> is for cross reference, <F2> enters the Tree module, <F3> shows the menu, <F4> produces a hardcopy, <F5> enters the Notes module.

New Commands include Markers, which allows you to 'mark' an individual in a number of ways and then use those marks to index, research, hardcopy and do many other

manipulative actions. This is a very powerful new feature.

Test is a most useful tool to enable you to check the integrity of your database. It checks for any anomaly within the tree structure. For example, it will highlight a person listed as both a mother and father or whether there is an unreasonable age gap between spouses.

I wish this had been a feature of the first program. I recall that in my enthusiasm to input data with the First Edition, my father was also shown to be my uncle! The Second Edition easily picked this error up, and it gave me 50 other warnings of possible errors. Most of these, fortunately, were just ancestors who had insufficient data entered against them.

In the Research Data section, we have many improvements. You can Print in lines and columns, Link and Unlink events, and even provide yourself with statistical data.

Full marks for the Geography facility. Now you can create and display a map showing the location of your ancestors. In addition, by using the Timespan function, you can step through time year by year to view the cumulative events in each place.

The tutorial section, which now covers four pages, helps you understand the workings of the program by taking you step by step through each of the features. This section is excellent and every minute spent following it will benefit the purchaser of the program one hundredfold.

I have really only skimmed the surface of the new features provided by this program. As well as all the things I have mentioned, Chris has provided a Conversion Service in which he will try and convert your *ARCHIVE* dbf files which you have created from the Microdrive Exchange *Family Tree* program into a format suitable for *QL Genealogist*.

### Easy use

Chris Boutal is an enthusiastic and accomplished family researcher. He is, however, sensitive to the needs of the complete beginner, so both his manual and program structure permit the most nervous user to achieve success with the minimum of skill and effort.

If you already possess the First Edition, then the upgrade is a must. If you have never looked into your family history before, then do buy the program, even the cut down version for the 128 K QL is first class.

```

Date: 12/10/90      Time: 1
QL GENEALOGIST - Integrity Check
V2.01 Sun 15 Dec 1991 17:15

7 has no birth year or sequence number
9 and 13 couple have an unlikely age difference
32 and 3 have an unlikely generation gap
32 and 4 have an unlikely generation gap
Child 32 born over 1 year after death of mother 4
Child 32 born over 1 year after death of father 3
32 has birth date earlier than death date
34 and 32 have an unlikely generation gap
33 and 32 couple have an unlikely age difference
35 and 32 have an unlikely generation gap
33 and 32 couple have an unlikely age difference
36 and 32 have an unlikely generation gap
33 and 32 couple have an unlikely age difference
39 has no birth year or sequence number
78 has no birth year or sequence number
79 has no birth year or sequence number
86 has no birth year or sequence number
81 has no birth year or sequence number
82 has no birth year or sequence number
    
```



# QL SCENE

## Hands-on QL now uses biofeedback

NESQLUG (New England Sinclair QL User Group) will demonstrate at their April meeting a new extension called PROD, developed in collaboration with the Computation Genetics Corp of Michigan, USA. The new program utilises a breakthrough bio-genetic thermoplastic (BGTP) developed by CGC.

The material has electrical conductivity properties which can be channelled to the comms (serial) port of a computer. The PROD extension interprets signals from a sensor pad made from BGTP. In conjunction with the extension, this permits heuristic interpretation of the user's needs when operating other computer software.

At a preview, CGC demonstrated how the user can operate a number of unfamiliar software titles without reference to manuals, by placing one hand on the pad, exercising concentration, and applying Body English. Remarkably, in 85% of attempts the user's desired effects were input to the QL. More startlingly, the operator, told that the program's function was word processing, formatted three paragraphs of justified word-wrapped text in Abacus as quickly as if it had been Quill.

### Sweat

The PROD extension currently takes account of ten parameters, including Sweat Salinity Factor, Body Language

Profile Index, left or right handedness, and Personality type (A or B).

### Byron

The CGC team is headed up by Arab-American computer high-achiever Dr R Ida bin Phulen, who is informally acknowledged to be descended from Victorian poet and all-round genius Lord Byron. "Initially, we selected the Sinclair QL because it was cheap," Dr bin Phulen admits. "Subsequently, attempting to transfer the concept to different computers, we found that these results only occur on an unexpanded QL with a JSU rom. Our software team suspects that PROD utilises extraneous code relating to the unfinished WHEN ERROR function on that rom. Rumours that only a British computer can respond to Body English are premature."

PROD will be released to the public domain as soon as the team are sure that the computer will not retaliate if it is input at too high an EOR (emotional overload rate). "We see this development as a major breakthrough in Artificial Intelligence. Our sole interest is the commercial application of the sensor pad," explained Dr bin Phulen.

It is expected that, when commercial release commences, the sensor pad will retail for about US\$400. A rumour that a 29-cent sticking plaster works just as well is being discounted by

## German journal

The Sinclair QL User Club eV has sent a copy of its German-language magazine, QUASAR. This is a stout, A4-size magazine of around 50 pages with a single-colour card cover, very well produced. For more information, write to **Franz Herrmann, Talstrasse 21, W-5460 Ockenfels, Germany.**



CGC. PROD and its source code will be freely available through NESQLUG.

Computation Genetics is expected to receive commercial and military grants of US\$75 million to develop BGTP applications on 1 April 1992.

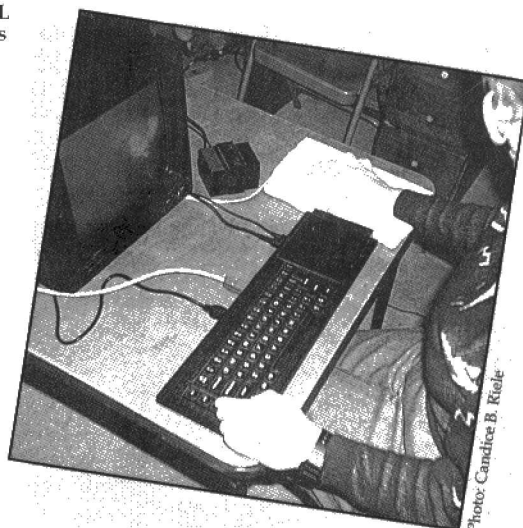
### Plaster

NESQLUG bulletin editor Peter R Hale was unavailable for comment. Colleagues believe that he is in bed with one

arm in sticking plaster after accidentally asking his QL for a cup of black coffee and a steak sandwich, 'well done', while carrying out tests in the small hours of the morning.

"We think the problem was," said a representative, "that it was after the mall closed for the night." Mr Hale is said to be "very surprised", but in no danger. All enquiries at this stage should be sent to the New England Sinclair QL User Group, PO Box 8763, Boston, MA 02114, USA.

A bio-QL advances on a test user during trials.





# OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide

somebody with the answer, or just sound off about something which bothers you, write to: Open Channel, *Sinclair QL World*, 116/120 Goswell Road, London EC1V 7QD.

## Thanks

Through your excellent magazine, I would like to thank reader Mr Frank West for supplying me with a copy of the 8056 printer manual, as requested in *Open Channel* of the January 1992 issue.

Anthony Landaw  
London NW9

*Editor's comment: our thanks to Mr West also.*

## Grumble

We the members of the Quanta Lancashire Area Sub-group be-

lieve that the location notified for the forthcoming EGM (Raine, Essex) will prevent the majority of members from attending. While we recognise that it is too late to alter the location of the EGM, we feel most strongly that holding the AGM in a similar location would prevent the views of many members from being expressed on a range of topics.

It is healthy and fruitful to hold workshops at various locations throughout the country, but the AGM should be held in one which encourages and enables the maximum number of members to attend – close to the geographical centre of the membership. Is this

not somewhere in the Midlands or near the north: Birmingham, Coventry, Manchester or Nottingham?

With Quanta's positive cash balance, the cost of finding a suitable venue should not be a problem.

The proxy system, too, worries us; it is hardly democratic for a blank vote to be handed to a nominee, whether the chairman or indeed any other Quanta member. For a member who wished to attend, but was prevented from doing so by distance, some facility for registering his vote is necessary, but there is no reason simply to transfer his vote to any other person.

On at least one recent occasion, the number of blank proxy votes given to the Chairman is believed to have been sufficient to swamp the votes cast by attending members. We believe this is not the wish of the membership.

Bill Barnett  
John Butterworth  
Mick Holland  
Steve Hutton  
Steve Johnson  
Peter Mangnall  
Roy Olive  
Mick Porter  
Keith Reader  
Joe Smith  
Mike Crossfield  
Quanta Lancashire Area  
Sub-group

*Editor's comment: This letter has also been sent by the Lancashire area subgroup to the Quanta committee. The location of important meetings is often a matter for debate within organisations with no fixed headquarters. The location is then most often determined by the willingness of local members to organise the event.*

*We have heard some interesting facts from the committee on this: they did a survey, and it turns out that the geographical centre of the UK in terms of Quanta membership is – wait for it – Luton. As it*

*happens, Raine, where the SGM (that's official) was held, is actually on the same latitude as Luton, although on a somewhat more easterly longitude. They did try for Northampton, but couldn't find anywhere reasonably cheap at short notice.*

*We have no definite news about where the AGM will be, but it seems pretty certain that it won't be Raine. Unless nobody else wants to organise it. . .*

## Characters

I have a problem with the character sets for the various channels. In the June 1986 edition of *QL World*, James Lucy wrote an article about this and mentioned addresses pointing to the start of the character fonts. He says that in the JS rom this was 163960 and . . . 64. I know from various sources that in the AH rom it was 167722 and . . . 26. I used to have an AH rom, and when I wrote print peek\_1 (167722) I got 41410 (start of char\$ block 32-127) or - with print peek\_L (167726) – I got 42286 (start of char\$ block 128 - 255). Then I got a Medic memory expansion and the start address shifted to 177258 and 177262 (still giving 41410 and 42286); I added something else, and the addresses shifted yet again, in jumps of 256 Kb to 176682/176686, 177290/177294, 177546/177550, 177802/177806. I presume that the addresses are different for the various channels, 0 for the input line and 1 and 2 for the screens.

Right! I now find that some games such as *Lock and Key*, will 'work' on my daughter's Medic QL using the address 177290 in line 2390, but not on my Trump Card QL, using this number, nor on either of the new numbers which I've now determined by hunting through the rom for those

## Editor's notebook

A distant rumble of thunder. The wind drops to nothing. The world waits. There are Miracles in the land (see page 16). What betides? Well, for a start, the name of Maxwell is now out of the equation. We have a new publisher – the event expected since November 1991 has happened quickly and painlessly. I came back from a short holiday and – ping! – there it was. No men with dark suits and New Zealand accents, just a tidy takeover. What this means for *QL World* in the long term I can't say. Flex your ears for omens. I shall be doing what everyone else is doing – watching the budget!

Talking of Miracles: we now have the Gold Card review in the can for the next issue.

Thank you for all the Grolley suggestions for the December issue. Nearly everything was pinned down by somebody, but nobody came up with the unobvious suggestion. Your letter page, a tasteful cream and violet confection, should have been the colour you see before you today.

Remember, if you can't get hold of someone in the QL business, and you can't get through to me, ask Quanta. They can usually help.



magic numbers 41410/42286. The relative addresses are now 180778/180782.

When I say that Lock and Key won't work, I mean that using address 180778 only gives me graphic char\$ to channel 0 (input lines) and not onto the screen. So my screens are full of the %\$\$\$:: char\$ and are not being affected by the udg, program.

I wonder if anyone there knows of a solution – in effect, how I can find out where the channel fonts are? In idiot language please!

**Dane Kurth  
Busswil  
Switzerland**

## Buttonholed

Disgrace! I shall have to give up *QL World* and subscribe to Tailor and Cutter instead! The button-hole of the jacket on your December issue is on the wrong side! Owzat for an 'extra Christmas present'?

Happy New Year!

**Ron Mustoe  
Chester**

PS: Don't worry, you haven't really lost a reader. Even if I don't win the prize. Thanks for a regular good read.

*Editor's comment: Yes, but it isn't really a mistake: you see, our artist had to copy his own reflection in the mirror in order to capture the full beauty of his profile. And I think he's made an extraordinarily good job of it, really, don't you? The trouble is – as a result – he now has both eyes on the same side of his head, and has had to apply for a transfer from the Surrealists to the Impressionists. Whether that is a mistake must be left, I think, to personal conscience. I prefer Botticelli myself. Happy New Year, but no banana.*

## DOS costs

There has been quite a lot of talk recently about the replacement of the QL. A point that emerges is that of cost. I get the impression that there is an expectation that any replacement will be a low-cost one, tending to establish the QL market as a low-end market more for hackers and enthusiasts than for

professional or business users.

Similarly, the expectation is that software will also be low-cost, that is to say, that around £100 for professional or business-type software is excessive or at the very least an upper limit.

I feel that if there is to be a real future for this market, users will have to realise that there is no such thing as a free lunch in computing. Good QL software is already very cheap compared with other systems, and one cannot expect software writers and suppliers to operate unless they can obtain some reasonable return for their efforts.

Anyone who has tried the alternatives such as disk-bound DOS-based machines will know how inconvenient, slow and awkward they are in use, despite their glossy appearance, so I feel that those who want to ensure a future for our technology should accept that at the outset, at any rate, the price will have to be paid to maintain and expand this.

**Frank Gutteridge  
Corsier  
Switzerland**

## Amend

Regarding P H Warne's comments (*Open Channel* February 1992) about my *Abacus* program for calculating building society income (OC October 1991), either he has misinterpreted my meaning of 'edit' or I have been as guilty as authors of some computer guides in leaving out steps in descriptions, assuming the reader was already one step ahead of me.

The point arises where I wrote: '... or one can go as far as March and then COPY the row March to get April...' But the sentence continues '... editing month-name and cell references where necessary.' This referred to entering a long formula into one cell and COPYING it to other cells.

Mr Warne is right in saying 'cell references in formulae are translated into relative positional references' when COPIED in *Abacus*. Therefore, in this case, one has to edit them. Perhaps I should have taken 'editing' a step further in describing my program in the October 1991 issue.

My method, once the row for March had been COPIED to get April, was:

1) To AMEND the row for April by enclosing it in quote signs (""). Thus all subsequent COPYING would be of text. (It is helpful with long formulae to widen the columns temporarily, so that all of the text is visible on the screen, rather than just see it in the edit window of *Abacus*.)

2) Once the rows in the grid had been filled in this fashion, I went back and used the AMEND command again, editing month-name and cell references where necessary, and removing the quote signs from the cells where text was intended to be turned back into formulae again.

With programs which have a lot of long-related formulae, I find this method less of a task than typing-in every single line completely. In any case, some hard work is inevitably required.

Incidentally, enclosing my formulae with quote marks is what I did to obtain the print-out for my first table, which appeared with my submission, and in my opinion is a better way of printing one's formulae to hard copy, than using the *Abacus* option contained in the PRINT command.

I was puzzled by P H Warne's suggested use of the dollar sign (\$) in *Abacus*. I have never come across this... my edition of *Abacus* will not accept it. I wonder how many readers have editions that will?

**Peter Tomlin  
London W2**

## Well done

Full marks for good customer service to Chris Boutal. Problems encountered with V2.02 of his excellent *QL Genealogist* running on my reliable AHrom QL were quickly solved and I received his update V2.06.

Chris identified that the Configurator file was not being loaded correctly – that is now fixed. Other things were associated with AHrom peculiarities, and without getting his own Am room from its loft, he fixed those too. Readers with AM and JM roms may like to note that a limit of 128 charac-

ters can be read with the INPUT command.

Some of my records were just over the limit, leading to a Buffer Full message and crash. Chris has rewritten the reading and writing routines in *QL Genealogist* to correct the error. Well done Chris, an excellent program and good customer service with it. My AHrom has been in daily use since 1985! It also purrs gently with *Perfection*.

**Peter Harrison  
Brislington  
Bristol**

## Toolkit 2

First may I compliment you on producing your New Users' Guide. CL CHIP sounds more like a culinary delight, though! The major problem I have had is in punching the holes to file the pages in my QL Users Guide. You have kindly indicated the location of the holes. Like a lot of people, I have a wo-hole punch which punches holes exa-

tly 80mm apart, just what is needed. In order to get the holes in the correct position, I need to locate the exact centre point between the holes – a tedious operation. Please would you print a centrin line between the printed hole positions? On an entirely different prob-

lem, I am using a Sandy Super board, version 1.6, which gives me 649K, disk interface and parallel printer port. Unfortunately, this version has only Toolkit 1. Plenty of suppliers offer Toolkit 2, but where can I obtain an upgrade from Toolkit 1 to Toolkit 2? Sandy no longer seems to advertise.

**E E Stocker  
Clanfield  
Hants**

*Editor's comment: We'll see if we can get the layout artist to put some dots in. Toolkit 1 was the original Sinclair program. Toolkit 2 was rewritten by Tony Tebby, and there is no 'upgrade path' between the two. To run Toolkit 2, you'll have to obtain it in 3.25 in disk, or eprom, and run it externally to the SuperQboard. Care Electronics sell Toolkit 2, and say, if you want information or help, phone them on 0923 672102.*



# QL

# S C E N E

## Digitiser into Europe

CL Systems have been in touch to tell European readers that the CQVI video digitiser is now available from **QLympic Computer Systems, Quellenweg 18, D-4220 Dinslaken, West Germany.**

Certain compatibility problems reported by users of the CQVI and Miracle Systems' Gold card are being looked into.

In the UK the CQVI is available straight from the makers. Orders and enquiries to **CL Systems at 403 Chapter Road, Dillis Hill, London NW2 5NG. Tel: 081 459 1351.**

## London Quanta in May

The London Quanta Sub-group are holding their third workshop on Saturday 30 May 1992, at St Helen's Church Hall, St Quintin Avenue, London W10, between 10am and 6pm. The Group will have a Help desk; a room for people to set up their own systems for demonstrations or tinkering; new and second-hand software and hardware traders, and QL personalities, very probably eminent ones.

For more information, contact **Jerry Davies on 081 863 1631** or **Colin Murphy on 071 328 5407**, or write to the **London Quanta Sub-group, 10a Lowfield Road, West Hampstead, London WN6 2PR.**

They are hoping to attract enthusiasts near and far. Don't disappoint them.

## Story links QLs to Cray computers!

**Robin Stevenson**

In December 1991, I reported in *QL World* on QL activity in the North American continent. It appears that a debate now reaching its third birthday, and mentioned in that article, continues to rage in North America. The hotly contested issue is the amount of truth in reports that Cray Research, builders of some of the world's biggest and most advanced computers, linked up a dozen or so QL motherboards to do research into parallel computing.

I suggested at the time that this report was 'apparently true'. Originally I was prepared to

believe that it was 'apparently untrue', but on the word of a US contact that he was confident of its truth, and that he had seen the photographs, I changed my mind.

After publication, however, another reader of *QL World* wrote to say that, as far as he was aware, there was more fiction than fact circulating. The story first sprang, he assures me, from a Boston Sinclair/Time Newsletters intended to be read in April 1989. Written anonymously, with no picture, it described the contraption, claiming it had been labelled the Cray Z-2B. You have to say that

quite fast, with an American 'zee'.

To the surprise of the perpetrators, the story then surfaced as a straight article in a number of other journals, creating, in the process, quite a few April Fools. Where the alleged photograph came in is anybody's guess, but the story refused to die. Whether there is more than that to this tangled web is hard to be sure. I now suspect that my original scepticism was nearer the truth. But life has an uncanny knack of paralleling art. Perhaps there is a Cray engineer out there who can tell us more.

## C68 V2.00

A major upgrade of the public domain C68 compiler has now been released. C68 2.00 comes with a full disk of documentation, extra utility programs, and an expanded C programming

tutorial. The floating point maths library has been overhauled, and memory allocation is more flexible. C68 is available from Qubbesoft, CGH Services, the Quanta library, and as an option with *Cport* from Digital Precision. Prices vary with source. Check which version is currently being supplied when ordering.

Computer Systems, Jochen Merz Software, Jurgen Falkenburg Computing, and Albin Hessler Software. Workshops and discussions planned include Forth, a new animation application, Quill, SMS2 Qdos-compatible operating system, Maulausch MausNet mailbox, the future of Qdos, QL networking, the public domain C68 compilation system, Fleet Tactical Command, and the latest Minerva rom.

Quanta is organising a mini-bus party. The exact location is the Schulzentrum/Realschule Munster-Roxel school, Lilbecker Strasse 24, Munster-Roxel. Munster is served by major road routes; from the Intercity railway station, the number 19 bus reaches the school at the end of its run; a 'white windmill without sails' is the landmark for the school in Lilbecker street. The nearest airport is Munster/Osnabruck, 20 km away.

More information is available from **Franz Herrmann, Talstrasse 21, W-5460 Ockenfels, Germany. Tel. +0(0)2644 1855.**

## Speedscreen

Simon Goodwin's Qdos screen accelerator *Speedscreen*, launched in *QL World* in 1987, has been re-launched with a £10 price cut, and the release of a working demonstration into the public domain.

The full version also accelerates the smaller text used by programs such as the Psion suite. It includes fonts, a character design task, and eight configurable versions to suit all systems. *Speedscreen* is available on disk or microcassette at £10. The fast rom version (with all the extras on disk or tape) plugs into the QL rom port, and costs £20.

Orders to **CGH Services, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA.**

## Germany

More details on the Sinclair QL User Club eV International QL Meeting in Munster, Germany on 21 March 1992. The Meeting will run from 8am on March 21 1992 to 10pm that same day. Set-up for demonstrators is possible from 4pm to 10pm the previous day (Friday 20 March). Clubs Sin QL air (Netherlands), SQLUG (Sweden), Quanta (Great Britain), QLItaly (Italy), QNV (Denmark) and NASA (Norway). Commercial exhibitors on the list by the end of January included Miracle Systems, IF Services, Dilwyn Jones Computing, PROGS Professional and Graphical Software, Qubbesoft P/D, Ingenieurburo Wilfried Krummery, QLYmpic



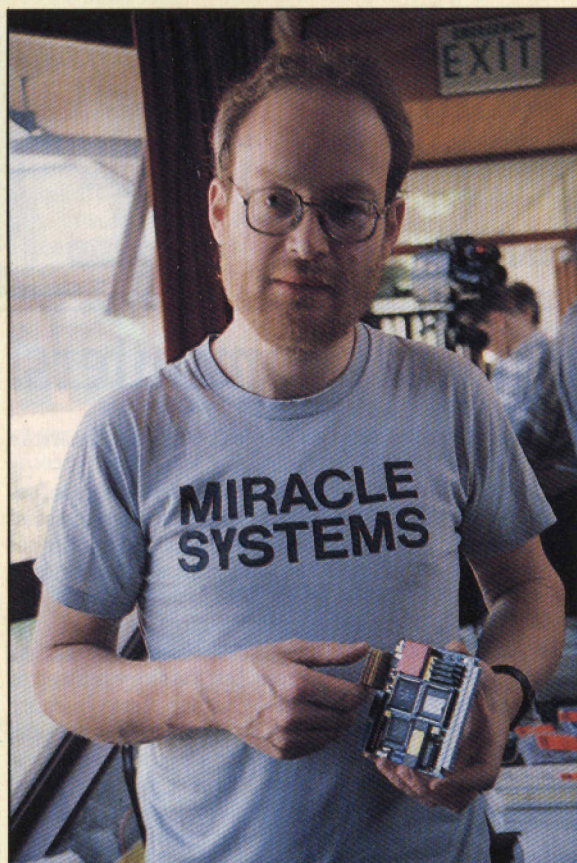


Photo: David Drysdale

# HITTING THE GOLD

**Miracle Systems are in their 10th year of producing hardware upgrades for the QL. Helen Armstrong talks to Stuart Honeyball about hits past and present.**

**S**tuart Honeyball is Miracle Systems. Not that Miracle is a one-man show; Stuart works with the best people in the field when he needs to, and has stayed on good terms with them. Miracle's high-performance expansion board, the Gold Card, has roused more interest among street-level QL users than anything since the Thor was announced.

Stuart is an engineer. After graduating, he worked for Ferranti, leaving after a year to go to Sinclair Research (before the time of the QL), where he worked on the pcb layout for the Sinclair Spectrum. He left Sinclair after nine months, tempted by a healthy rise, only to see his new employer go out of business. With some savings and no job, he set up his own business.

"If they hadn't gone bust, I wouldn't have gone self employed," he says, like many business people looking back on their origins. Miracle's first major product was an RGB interface for the Spectrum. Like many a first stab, it probably cost its inventor money, but it taught him how to run the business. In December 1983 the QL was announced, and Stuart saw a glaring opportunity straight away.

"I immediately noticed that it didn't have a Centronics port," he says. "It seemed logical that there was a pretty large need for a serial to parallel converter, so that the serial ports on the QL, which are a bit specific to the QL, could be used to connect the QL to standard printers. We designed the QL Centronics Interface to do the job with the minimum of fuss. Oh yes! We're still selling them." The converter has

a very low power 74HC164 chip at the heart of it. "It's a nice, simple device. Serial cards for printers cost around £60 or £70 then. Some were over £100. Setting up the RS232 system is pretty tricky, as well. We started at £49, and with Centronics there is no setting up to do. They really made life easy for people. We sell them for £29 now."

Getting information about equipment was far from easy. "We had to do most of it by intuition – you put yourself in the position of the designer, and guess what he did! It's stood us in reasonably good stead. That was our first major product – we have sold about 15,000 of them. Other people copied the design but we reckoned that, if we could come up with it in the first place, we were good enough to improve it. So we came up with a dual port one – the two serial ports are wired differently. Necessity is the mother of invention!"

## Expanderam

The next important product was the Expanderam, their first occupant of the QL's expansion port. CST already had a disk interface on the market; others followed, and ram expansion cards began to appear. Miracle's ram expansion, however, had an innovation – a through-connector. So was born the legend of the long QL! "So now they were buying their ram expansion from us!" The Expanderam also broke ground by implementing its memory controller in low-power HC chips, keeping the current demand on the QL (which can

deliver a maximum of 0.5 amps) low enough to run both an expansion card and a disk interface.

Obviously, the next logical move was to incorporate the disk interface. The Trump Card was born, outrunning other disk interface/rams by using the entire QL address space. The maximised memory brought the Trump Card to 768K, and the QL to 896K. The device's programmable rom has the disk drive software and *Toolkit 2* on it. All the software on their card-based products is written by Tony Tebby. Successful despite its protuberant outline, it became a legend in its own lifetime.

Stuart sensed that memory was more important to users than peripheral space. The software of the QL is set up to run 16 peripheral cards, but the hardware doesn't have the space for it. "If you put more and more on one card, people are quite happy with that one card," he has found.

The newest 'mainstream' product is the Gold Card, marketed with loving care in its gold-patterned box. The Gold Card isn't entrusted to common black carbon-impregnated antistat bag – because the contact would run out the battery of the battery-backed clock. "We didn't think about that until we were just about to send them out! Would have made the clock a bit of a waste of time."

So what does the Gold Card do that nothing else does? "Quite a bit really," he says, in his understated way. His modest assessment is that, with the 16 MHz 68000 processor, programs run about four times their usual speed. "People notice that.



We've had a lot of nice letters about it. It makes a big difference to programs like *Quill* and *SuperBasic*." The Gold Card operates by copying the contents of the QL rom to its own 2 MB on-board ram.

So he's built a QL on a card? "That's right – it does everything apart from the I/O and the housing." The screen has been speeded up by shadowing, so that write cycles go to the QL to update the screen, but read cycles come from the fast ram. The disk interface section uses a new disk controller, to handle the same disks as the Trump Card (at 720 K), as well as the new breed of 3.2 floppies. It exploits double density (DD at 720K), high density (HD at 1.44 MB) and extra high density disks (ED at 3.2 MB). Setting their own standards, Miracle eschewed the PC standard 2.88 MB ED disks. "The QL is not a PC, so it seemed a bit daft to settle for 2.88 when we could have 3.2."

## Four-layer board

Like the Trump Card, the Gold Card has no through-connector. "It would have been very difficult. But we have more complaints about the Trump Card sticking out than we do about either of them not being able to address other cards!" The Gold Card boasts a meticulous hand-laid, four-layer motherboard, carrying (among other things) the 68000/16 processor, four 4Mbit drams, the 8473 disk controller, and the message 'Hi Dawn! Dawn is Mike Tomlinson's wife – Mike laid out the board. Who says that romance is dead? "By putting a Gold Card in, you can stop the QL crashing," adds Stuart, "because the processing takes place on that board, rather than in the QL."

So you don't need a QL to run the QL any more? Just a boot rom? "And some I/O,

things and dropped things. We didn't want to put the price up or make it larger."

The Gold Card has done well. Launched in May 1991, it has sold twice what was expected. There has been a groundswell of enthusiasm for it, even among those who don't have one. Some people have tried the PC, Stuart reckons, and found it doesn't do what they want it to do. The QL

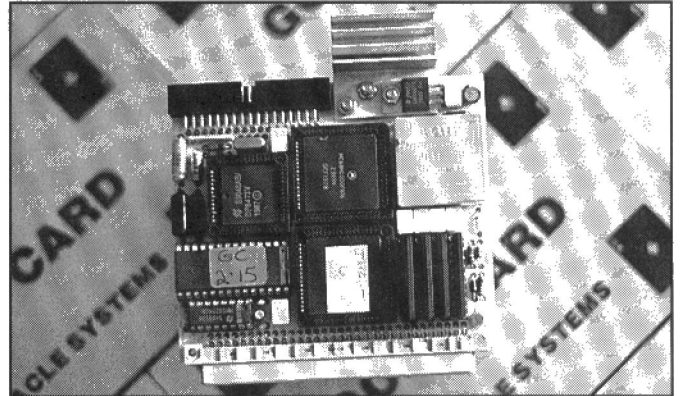
**Right: Gold Card... Below: peeking out of the QL.**

is friendly to use, and it does true multitasking, which the pc doesn't seem to be able to do it properly, even under Windows 3. My engineering consultant agrees.

"We've made enough out of the GC to persuade us to go along and do the next logical thing," says Stuart seriously. "There are two things that need addressing: the graphics need improving, and we really want to replace the QL hardware. Fax cards are "one thing we have in hand, but there's a lot more than that. If everybody went over to buying a QL or a QL compatible," he adds, "it would create a lot of British jobs."

Stuart travels to Europe with his bike to drum up overseas orders. "If they don't see you, they won't buy," he says. "They don't want to risk their money." He laughingly describes the horrors of cycling across the vast, cobbled Place de la Bastille, with Parisian drivers whizzing crazily around

Mild-mannered Honeyball is also offended by banks, and stays away from their loan facilities as much as possible. "It's only the big four which are a problem," he says. They bank with the TSB, getting interest on the account. "It's people who have personal accounts with those banks who enable them to go on working in an inefficient manner. You can get cheque

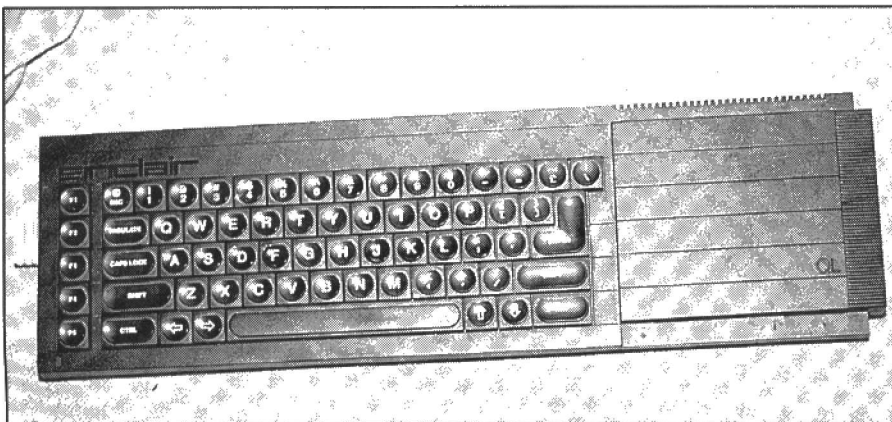


accounts from several Building Societies." People are listening carefully to this sort of talk these days. I comment that it doesn't take much, at the moment, to make people twitchy. If anyone is out of touch for a while, *QL World* gets letters and calls. More than we used to.

I put to Stuart the claims occasionally voiced that Miracle have ignored the work of other developers. "We have incorporated some of Lawrence Reeves' code, at his request, to make the Gold Card Minerva compatible. My conscience is clear on that point!" he says. "Wherever we can, we have changed things, which has involved us in quite a lot of work. We didn't design the Gold Card to be part of an expansion system; but we take all comments on board, and we attempt with each new product to make it compatible with as much as possible." Subject, of course, to meeting their original design criteria. Stuart attributes much of his stability to knowing what the market wants.

## Indignation

The notion – lobbed at not a few designers over the years – that they designed their master product to be difficult for other QL designers to follow through, rouses nearly as much indignation as the banks. "We don't do anything cynical like that at all," he says promptly. "That is completely against our philosophy. Where we have cut off avenues, it's either where we just haven't thought about it, or where it enables us to provide more in that particular product. On the Trump Card, we went for 768K, but you can't add other expansion cards. The vast majority of people prefer the extra memory. We don't envisage people using the Gold Card with other expansion cards – it fits neatly inside the QL." It's hard to please everyone. Games players



and a few other things," says Stuart. "We're thinking about something along those lines, but we can't say exactly what at the moment. If we'd talked about the Gold Card before it came out, it would have been completely different from what we'd told you. We got quite a long way down the design of one gold card, then we scrapped it and started again. We realised that we could get it all inside the QL. We added

them. He takes his bike across Europe on the train, and then cycles. That's a long journey. A short journey is Nottingham – 80 miles each way, with his monitor and rig on his back. "You can get quite a lot on a bike," he says earnestly. He used to have a flash car, but he worked out what it cost to keep it standing in the drive, was offended, and sold it. He makes a good case for rail transport.



## MIRACLE SYSTEMS

soon told him that the Gold Card went too fast. "Now we have a SLUG command. You can slow the card down to any speed you please!"

### Design balance

I sense a fine judgement balancing the benefits of building their own standards, and meeting the criteria of other designers. Their success has been their survival: one of the few businesses which supports itself — development, contract work like Tony Tebby's software, Stuart and his full-time employee Mike — entirely in the QL market. You can only do that on reliability, trust and repeat custom. Miracle also offers a 14-day money-back-if-not-entirely-satisfied guarantee on all its products — "I don't know why more companies don't do it. It sets people's minds at rest, and we only get back about one a year." And two to three day delivery — "We find the post is reliable, but if it doesn't work well, you can use a carrier."

They have worked to make the Gold Card compatible with as much software as possible without moving away from their original concept. "We've got plenty of space inside the rom — we can keep on adding patches, but it does take a lot of work." All the software on Miracles card products is written by Qdos author Tony Tebby. "All we do is pass comments to him and ask him if he can get it working with this, or that." Tony is actively working with Miracle, although he is seen about the QL community less these days than of yore. "He's a good bloke, though, old Tony. We pay our bills. A lot of people have been ripped off in the QL business. If we say we'll pay for things, we do, even if we can't use them in the end." It's a policy which has worked for them.

Miracle have made their mark with other products over the years — the QL Modem. "Probably the product I'm most proud of! I'm pretty sure it was the smallest one in the world when it came out. It uses optocouplers instead of an isolating transformer." The modem is no longer on sale, but it was a best seller in its day. Miracle's

Midi board, on the other hand, was only adequate." The software wasn't what people wanted. I gave the MIDI to Quanta in the end. It was a fair bit of work to churn out, and we want to concentrate our resources on the high end stuff."

Miracle also produced a hard disk drive, now overshadowed by the Gold Card. "We find people are happy with 3.2 MB disks — since the Gold Card we've sold fewer hard disks, because the floppies are now big enough. PC software tends to be verbose. QL Software is concise, so hard disks are less necessary. We sold a respectable number, but it's not a product I particularly like. Some users require large storage, and bulletin boards do, but it's not really a mainstream item."

### Rebuilding the QL

What he really wants to do is rebuild the QL. "The operating system is the principle advantage of the QL — the soul of the computer. It was released before the bugs had been got out. Tony Tebby wrote it in about three weeks. He has to be congratulated on writing such a sophisticated operating system in such a short time! He's rewritten the operating system now. He calls it SMS2. It's likely to appear in one of our products."

There will be an advanced QL 'sometime this year'. Deciding what to include takes most of the time. You have to weigh things up. It must be what people want, at a price they are prepared to pay, in a time scale that we can afford to do it. We don't want to go away from standards if there's

no advantage. The keyboard in any product would simply be a PC keyboard, and we would provide a pc keyboard interface. We'd put a Centronics port in. We don't want to get tangled up in the commodity end of things. We want to provide what people can't get anywhere else." The idea behind such a 'new' QL will be to run QL programs much faster

**Right: Cornered! Mike Tomlinson works for Miracle full time. Below: Old friends. The Trump Card settles down with a good book.**



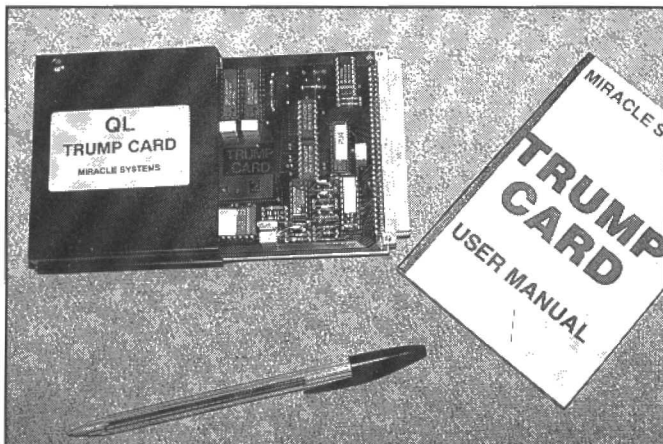
than they currently run. Programs which need code from a microdrive will be a problem, and may not be implemented until someone writes a patch. "That's getting rather close to piracy, and we don't want anything to do with piracy." They expect to be able to use the 68030 and 68040 in due course. "They are faster and better than the 68020." There are various clock rates available. "We will always give the best value for money basically."

There's a lot of thinking involved. While

### Test rigs

developing the Gold Card, they rigged up a 68020 to test. "We were extremely disappointed with the performance. The 68020 at the same clock rate as the 68000 runs 1.5 times as fast! If we had used an expensive 68020 with a high clock rate, it would run the Gold Card twice as fast, but double the price. I don't think that's economic."

It's as well Miracle take such trouble, because it proves that their faith isn't blind. "I think the outlook for the QL is quite good. We've gone so far down the line, and we don't want to leave the market," he says. "And I personally don't feel I've finished my mission yet. There's still a few things to do." And so say all of us.





# SOFTWARE FILE

## INFORMATION

**Program:** *Picturemaster\_Plus*

### Supplier:

Dilwyn Jones Computing  
41 Bro Emrys  
Tal-y-Bont  
Bangor  
Gwynedd  
LL57 3YT

**Price:** £24.00. 3.5 or 5.25 disks only.

Requires at least 256K memory. A monitor should be used.

## John Shaw gets a pictorial update.

Joe Hafke is something of a perfectionist. He is always searching for better ways of improving the product. Such is the case with *Picturemaster\_Plus*.

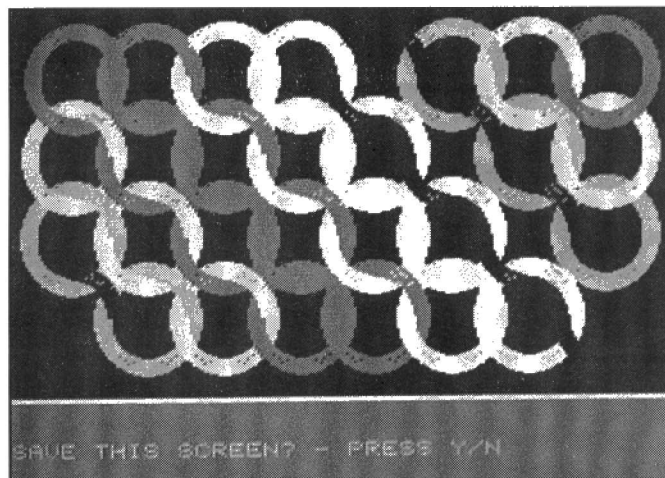
It has much in common with the original *Picturemaster*, which I reviewed in the November 1991 issue of *QL World*. You can now use very large text (up to eight times CSIZE 2, 1 in height). There is the addition of triangles to drawing tools (single, or patterns). Paper and ink colours in Text Window can be controlled. Part of a screen may be magnified.

The screen picture can be repeated in condensed form. There is full customisation of drive, printer and sound defaults, and a combined Fill and Pattern facility.

You can now adjust the screen image position, Store and Blend Screens. Handwriting can be carried out directly on to the image. There are now 15 more pre-designed screens.

Should you require it, you can generate a continuous show of the 45 pre-designed screens.

A very useful bonus, particularly to those intending to use it for advertising, etc., are the three lines of auto-centralised text which are now available.



## Picturemaster\_Plus

With regards to the screen painting, Joe has introduced a novel concept by allowing users to have at their disposal a number of Pads, with different shape painting faces, such as a line, a box, an ellipse (including a circle) and a triangle. Each of these Pads consists of two parts: the outer, narrow rim and the inner core. If only the outer rim is used, the single line outline of the figure appears on the screen, but if the figure is to be filled then both the inner and the outer parts of the Pad can be employed together.

Any painting colour within the range of colours available for the screen MODE chosen, including the Stipple colours, can be used and you can consult the full QL colour chart, if you wish, each time you are requested to choose a colour.

No drawing, text printing, or erasure is ever irrevocably committed. The latest effort can always be 'undone', without affecting the work previously invested in making up the screen.

The Handwriting is really the use of a small Box (rectangle) with automatic Fill and Leave Trails, although the size of this

Pen Box can be increased and the box rotated, like any other Box (but probably to little advantage).

An additional Handwriting facility is an optional grid of horizontal lines (like lined stationery) to keep the writing in a straight line.

Another *Picturemaster\_Plus* facility is the option to Move the whole Screen picture if it appears to be growing lopsided as detail is added (a frequent experience of mine). In this operation it is possible to lose parts of the picture by Scrolling, or Panning over the edge of the Window (WINDOW #1).

As an aid, a temporary line in a contrasting colour is displayed at all the four edges of the Window and remains in that position however the screen picture is moved.

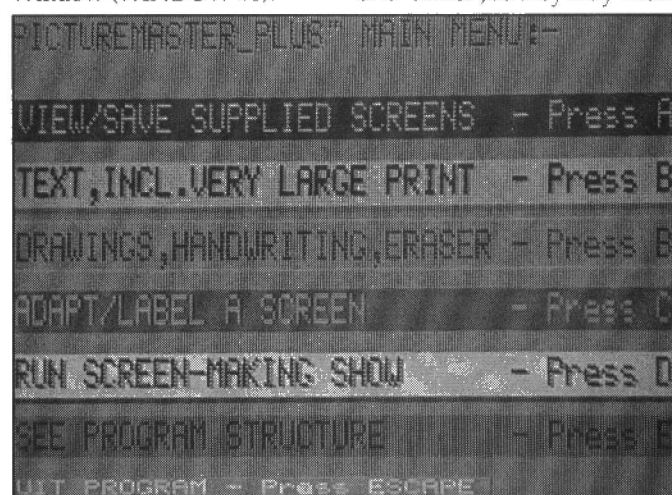
Whereas most of the text printing facilities are similar to those contained in the earlier program, a major addition is that of Very Large Print. This enables printing of Shadowed text of up to eight times the height of QL CSIZE 2, 1, with the character thickness 'fudged' to keep it in readable proportion to the height.

## Blending

*Picturemaster\_Plus* has a memory 'Store', large enough for a complete screen. A Stored screen can be Blended with the currently displayed screen. Blending produces a combination of the two screens, each on alternate horizontal pixel line.

Selecting this option starts a continuous, dynamic display of all the 45 pre-designed screens being generated. It is one of the principle features of this program that the user is guided at all times without ever losing sight of the Screen being designed.

As in Joe's previous work, the bulk of the program is supplied in fast loading Basic code to enable users to adapt it and 'tinker', as they may wish.





# HALF A LAPTOP

**David Drysdale finds the Z88 is the ideal driver for his portable printer.**

I decided years ago that my laptop, when I got round to buying one, would be a PC. It would assuage that nagging feeling that somehow I was being sidelined by sticking solely to the QL, and there were so many PC models to choose from and so many programs around.

That was the idea, of course, but the reality is that I have very little use for new programs. My QL is a workhorse with a daisywheel printer for its 'cart' and, together, they are ideally suited for a freelance journalist.

Word processing is the main requirement of most self-employed people, and good old *Quill* has been a mainstay. It is slow, and pretty inept on very long documents, but it gives little trouble with my stories which seldom run to more than 2,000 words.

However, one hiccup to my working system occurs because I am often away from home – sometimes for eight weeks at a time – cruising my canal boat, and work needs to be carried out during this period. The last time I set out to cruise, an editor rang just as I was leaving, asking for a long article, with pix, on a subject I had never

**The 'office in a briefcase' – the Z88 and a Brother EP44 typewriter.**



**The author's floating office – the canal boat Pegasus.**

tackled before. The deadline was in ten days' time – holiday or not.

Situations like this mean using my Brother EP44 electronic typewriter, but any typewriter is a word processor. The EP44, however, has a couple of features that started me thinking again about a laptop.

Firstly, this typewriter has a serial port and doubles up as a battery-driven portable printer, ideal to link up to a laptop. Secondly, it can download a paragraph or two of text matter directly to a QL, and this showed me how useful a laptop with the capacity to download a story would be.

The ability to transfer text matter to the QL was hampered by the fact that, in *Quill*, the documents acted like lumps of concrete and wouldn't reformat without my painstakingly removing surplus spaces, etc. line by line.

Then Digital Precision's *Perfection* word processor came along, and I was able to turn those chunks of concrete into pliable putty. I found I could work well by typing out stories in odd moments and downloading to the QL for making up and printing, but the memory limitations of the EP44 soon became burdensome. Not a real computer, it was only 'half a laptop', and something better was needed.

The phrase 'half a laptop' kept running through my mind. Because my work was being downloaded into QL with a colour monitor, PC-type keyboard, memory upgrade and twin disc drives, there was no need to duplicate these facilities. Half a laptop was what I needed – the keyboard.

A glance through the catalogues soon picked out a machine – the Cambridge Z88. It was one of the least expensive, far smaller and lighter than any of the others and – not having a full-size screen or disc drive – could operate on four pen-torch batteries. Although the machine fitted into

my 'half a laptop' category it is, in fact, a computer with a built-in word processor, spreadsheet, database and diary.

This full specification made the computer ideal. By linking it to the Brother EP44 typewriter/printer, I had full wordprocessing facilities and could tackle any editorial assignment from the canal boat. The only drawback was that memory could be very expensive in comparison to disc storage. A one-megabyte ram pack, for example, could cost £215, and a 250K plug-in eprom, the Z88's only system of built-in permanent storage, costs around £65.

From a half-a-laptop point of view, this didn't matter. If the data could be downloaded into the QL and saved on disc, the problem was solved. A 128K ram pack and a 250K eprom pack, costing £103 between them, give ample storage for the periods on the boat.

A commercial pack for linking the Z88 to the QL can be purchased – as can all Z88 equipment – from TK Computerware. In the event I had a cable made up by Adman Services of Telford, and sent for Phil Borman's freebie QL-Z88 link program from Quanta. This program disc, Comms 1, also gives cable-wiring instructions.

The first piece successfully transferred by this system was a file from the Z88's *Pipedream* database. It was received in no fit form to be used on the QL, but when transferred to the Z88 again it loaded as a perfect *Pipedream* file, solving the problem of inexpensive storage of Z88 files.

The next test – the crucial one – was how the Z88 would transmit text files, which would have to be imported into the QL in a form that could be used for word processing. The Quanta program was designed to transfer text files acceptable to *Quill*, and this it did successfully using the 'Import line by line' facility. The text, however, would still not reformat in the same way as *Quill*-produced files, and was not ideal.



The Z88 linked to the extended QL; the QL itself is tucked away beneath the wooden plinth.

Perhaps there is some facility in Quill for overcoming this drawback, but I looked in vain for it.

The next step was to try the same thing with *Perfection*, and here it loaded in an entirely different form. There is no 'import line by line' facility in this software. The file was loaded in the conventional way, but it came into the QL in one continuous line completely filling the line buffer. Only part of the document could be loaded.

## Reformatting

One answer to this was to reconfigure the program line buffer, but there is also a facility within *Perfection* to split extra-long lines which would have the same effect. Once this was sorted out, the reformat text command rapidly knocked the document into shape.

End of line codes were also visible on the screen, but the search and replace facility which, in this program has the power to remove embedded codes, soon cleared them out. Success at last. The two halves had come together to give me all the facilities I needed.

There is a limit to what can be accomplished, of course. It is not possible with the Quanta software to convert Z88

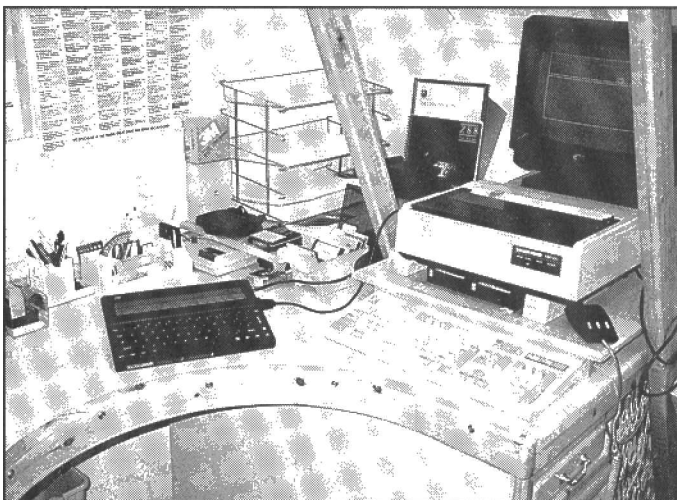
*Pipedream* files into QL *Archive*, for example, but the Z88 is a fully fledged computer in its own right, with notebook facilities such as a computerised diary.

Having used a paper diary for years, I looked on the idea of a computer based diary with suspicion and, sure enough, having spent some time making entries to the Z88, the whole thing was wiped out of memory when I plugged the machine into a mains transformer. This is apparently a weakness of the Z88, although, according to the handbook, it is rare for files to be lost in this way. The next model of the Z88, due out in the Spring, will have a separate battery backup for the rampacks to prevent this happening.

Perhaps the most useful feature of the diary is its ability to search, list and print. If a record was needed, say, of letters posted over a period of time the computer would

search them out and provide a printed list, complete with dates, provided that the items had been linked with a keyword, such as 'posted', for the program to search out.

The Z88, of course, will not suit everybody, particularly those wanting a full sized screen instead of the Z88's half a dozen lines, but for most of the time I don't even look at the screen. I just churn out the words, and sort the whole thing out when it reaches the QL. For anybody wanting half a laptop that will do a full-sized job, I can thoroughly recommend the Z88.



# EASYPTR II

Development System for the Pointer Environment

## Menu generator EASYMENU

Design menus on the screen

## Sprite generator EASYSPRITE

Design sprites, blobs and patterns on the screen

## SuperBASIC Toolkits EASYMEN/EASYPTR

Menu and pointer control in SuperBASIC programs

## Assembler source code generator EASYSOURCE

Create commented menu definition source code

**Budget version £33.00**

**Full version £49.00**

**DISA**

**Interactive disassembler £29.00**

**Export prices excl. VAT and incl. p&p (Europe only)**

**Albin Hessler Software • Im Zeilfeld 25 • D-7447 Aichtal • Tel. ( ) 7127/56280**



# ONE MANS

## S • Y • S • T • E • M



**F**or the last nine months, I have been using an Atari Mega 2 with one of Jochen Merz's QL emulator boards in it. I have no hesitation in saying that it is superb, and would recommend it to anyone with the money to spare. Note that it will *not* work on an STE.

In detail, my system consists of Jochen's own Extended4 emulator board and accompanying software, together with a Philips colour monitor and a Power Computing 40 MB scsi hard drive. Although the total outlay amounted to about £1200 (£400 for the hard disk, close to £200 for the emulator once I'd paid the VAT, and £600 for the computer and monitor, which I bought second-hand through the pages of MicroMart), and I hesitated for at least 18 months before making the move, I have no regrets at all.

I fitted Jochen's board myself, having sent off for it by post and having had to pay VAT to the postman on delivery. In principle, the fitting was not difficult except for one operation, but it took some nerve, and I am used to soldering. Six or seven wires had to be soldered to the mother board, which was easier than I feared, but one had to be soldered to the glue chip, and removing the chip from its socket to do the soldering was difficult and scary. The design of the chip itself also means that when you push it back into its socket, it is easy for short-circuits to be made inadvertently between the corner legs on the chip. Fortunately, the chip is robust, and does not seem to have been damaged. Cutting two of the tracks on the mother board was also an alarming first for me, but in principle straightforward, if crude! On reflection, I would probably ask Jochen to install the board if I had it to do again, although this would involve horrendous postage costs.

The software that comes with the emulator includes among other things a first-class transfer utility for reading from and writing to MS-DOS disks. I use it all the time.

Once installed, the emulator is a dream to use. I paid out the few extra pounds for Qdos on Rom, so that the machine boots straight up into Qdos on power up or reset. From there on, it is a very fast, totally reliable QL with massive memory. The one program I had to change was the Psion printer driver, to take account of the Atari's parallel printer port – which also needed a proper parallel cable, of course.

My use of the computer means that

**Hugh de Saram is an Atari/QL emulator enthusiast.**

multi-tasking is an absolute first priority, since I switch constantly between word-processing and database applications, together with occasional spreadsheet work. Ideally, I need three databases running all the time, so even with a 768 k Trump Card the QL is inadequate. Miracle's new Gold Card would be an alternative now, but the Atari is, I feel, both more elegant and more durable as a piece of hardware. The Mega keyboard is particularly good. So I use *Qpac II*, and run one copy of *TurboQuill+* and three of *Archive*, together with some of the utilities from *Qpac I* – calculator, calendar, and above all the configurable clock/calendar which I can pop up with the current date (always correct, since the Mega has a battery-backed clock built in), and then paste straight into my *Quill* letters by pressing ENTER.

I have no call for graphics applications, so I cannot comment on the compatibility of QL graphics programs with the Atari hardware; Jochen would be able to answer any questions over the phone. He was very patient with me as I went about fitting his emulator board and speaks excellent English!

Adding a hard disk to the Atari makes the machine a real power house. The scsi format makes for extremely fast data transfer, and allows more than one hard drive to be daisy-chained from the computer's one hard-drive port. I also find that, once everything is booted up, it is perfectly all right to switch the hard-disk off and on as needed, while leaving the rest of the computer running: my particular drive unit is not quite as whisper quiet as it is advertised to be, despite not having a fan, so I only flip it on when I actually need it.

One modification I have made is to the keyboard cable. I wanted all the computing clobber to be on a rack of shelves beside my desk, so as to leave the actual desktop clear except for the keyboard and monitor. So I went out to Tandy's and bought a six-wire telephone extension cable, and then to a local computer bits-and-pieces shop for the little plug units for each end of the cable, and made my-

self a two-metre keyboard cable. Now that there is a DIY tower box available for the Atari, designed to stand on the floor beside your desk, you can actually buy a ready-made long keyboard cable.

The speed of the Atari means that Quill can no longer be accused of being sluggish, and with the enhancements provided by *TurboQuill+* – the glossary function in particular – it becomes the excellent budget WP it has always been for the MS-DOS system – yet without the document length limitation which that system so annoyingly imposes.

*Archive* outperforms its MS-DOS brother on XT hardware, although *QL-Archive* lacks a couple of features of the MS-DOS version. File sorting is especially fast, somewhat making up for the lack of index files.

If I ever feel that, with the move up in the MS-DOS world to 286- and 386-based machines, the Atari/QL has become unacceptably slow, there are a number of accelerator boards now available, from a 16 MHz upgrade of the 68000 chip, to 68020 and 68030 replacements. Whether the last two work with the emulator, I couldn't say, but the first one does.

Equally, although for the moment 2 MB of Ram is quite adequate for my needs, there would be no difficulty in having another two installed. And if I were willing to put out quite a lot more cash, I could replace the Memory Management Unit and award myself the capability of increasing memory up to about 16 MB!

Finally, the greatest attraction of the Atari/QL is that it is on this machine that Tony Tebby seems to be concentrating his software development efforts. What other argument do you need?

### INFORMATION

QL Extended 4 Emulator £163 + VAT  
Jochen Merz Software  
Im Stillen Winkel 12  
4100 Duisburg II  
Germany.  
Tel: 010 49 203 501274.

Power Computing Slimline 40MB scsi  
hard disk £425  
Power Computing Ltd.  
Power House  
44A Stanley Street  
Bedford MK4 7RW  
Tel: 0234 273 000.



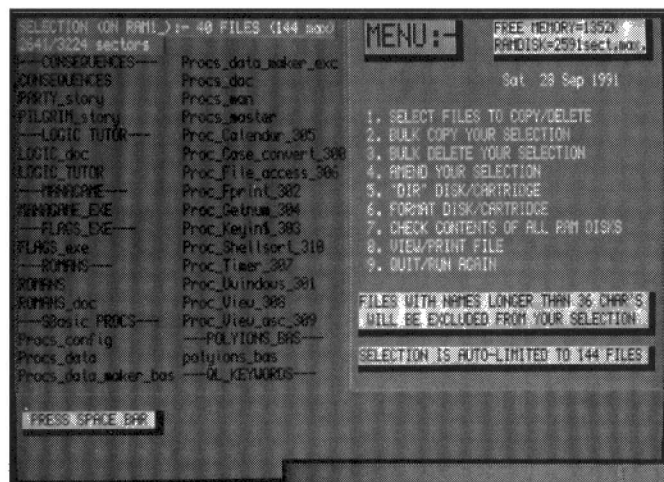
# SOFTWARE FILE

## INFORMATION

**Program:** *Filemaster*

**Supplier:** Dilwyn Jones  
Computing  
41 Bro Emrys  
Tal-y-Bont  
Bangor  
Gwynedd

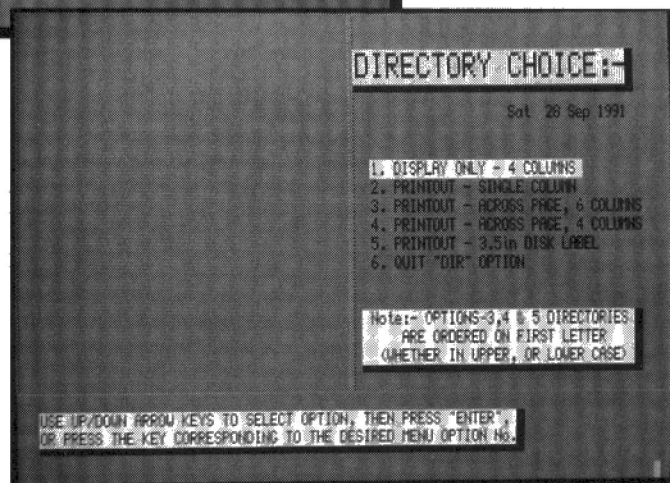
**Price:** £12.00. Mdv, 3.5in  
or 5.25in disks.  
Expanded QL  
with at least 256K  
extra memory and  
a ramdisk system.



**John Shaw**  
looks at a new  
file handler  
designed to do  
everything.

**W**hy write another file handling program when there are already many such programs on the market? asks author Joe Haftke in the opening paragraph of the 13-page Manual which comes with this utility.

He answers his own question by saying: 'There are many excellent file handling programs in existence, but none of them performs all of the tasks which are necessary to look



# FILEMASTER

after a system of files.'

Does this one? Well, you be the judge. The *Filemaster* capabilities are listed as:

Batch copying of up to 144 files to one or more medium. The selection needs only to be made once and can be assembled from several different media.

Deletes redundant files from different media using a single selection of all redundant files from all the media involved.

Formats cartridges and disks.

Displays media Directories of up to 72 files per screen in four columns without scrolling.

Prints ordered directories with date across the page (250 per A4 size page provided your

printer can use condensed script).

Prints ordered directories of up to 50 files on a 3.5in disk label. (Printer must be able to handle Superscript.)

Prints both the above directories, grouping together files beginning with the same letter (upper or lower case).

Files can be viewed on screen or printer.

Can it be used with your system?

Well, this program requires the availability of an already active ramdisk system catering for eight ramdisks. It will however run from microdrive, floppy disk or hard disk. A monitor is obligatory.

An expansion of 256K is essential, however, one of 512K or over is a decided advantage, as the free memory decides the number of files that can be handled in a single selection. I estimate that a *Trump Card* will cope with the bulk handling of about two thirds of a 1440 sector floppy disk in one selection. I can confirm that a *Gold Card* will hold the entire contents.

In essence you have a file system which collects all files from your chosen sources and puts them into a ramdisk from where they can be manipulated. They are then downloaded to whatever media you choose with good speed and efficiency.

The whole procedure is menu driven and error trapped.

The program is provided with an easy-to-understand manual and a large section devoted to make sure that your particular printer can be configured to produce the necessary size of print to use on the disk labels.

To summarise, *Filemaster* is a well written utility, providing a comprehensive ability to arrange, rearrange, merge and print all your files. An asset to all QL users, an absolute must to all owners of hard disks and, of course, those fortunate people intending to buy the new Miracle Systems 3.2 megabyte disk drives.



A word processor is a very personal thing. It takes a while to get used to it; then you find that you can't do without it. When a new one comes along, you find that you can't get into it because of the familiarity of the old one. Your attachment amounts almost to an addiction, that is the only way to explain why so many people still use *Quill* (apart from the fact that you have to pay for anything different).

Quill is adequate for producing a short document, but it has difficulty handling long documents. Its range of operations is restricted. Its major defect is that it is slow enough to drive you mad if you need a word processor frequently for reasonably long pieces of writing. The problems with Quill have led to the production of a steady stream of alternatives.

The field is so rich that I'm going to restrict myself to a few of the many programs which could conceivably be used for word processing. I shall leave out text editors in which printout is a secondary consideration, so I shall ignore *Spy*, *Master Spy*, *QD* and *The Editor* in its ordinary version. I shall also avoid desktop publishing programs, which provide methods of laying out pages of text and graphics, and in which it is often easier to read in text from a file created by another program. This means ignoring another range of excellent programs – *Front Page*, *page Designer*, and the three (!) DP packages – *Desktop Publisher*, *Special Desktop Publisher*, and *Professional Publisher*. I shall also ignore other programs capable of doubling as a word processor – notably the excellent *FlashBack*.

### Meet the Editor

What are we left with? *The Editor Special Edition* (which unlike its ordinary sister has a proper printer driver), *text87*, and *Perfection*. It could be that price is a major determining factor in the choice of program. For the versions I shall deal with, the current prices are *The Editor Special Edition* 2.05 nearly £50; for *text87* 3.01 £60 (though recently I have seen it advertised only with companion programs for nearly £95); for *Perfection* 2.00 nearly £80. Note that the price of *text87* includes as standard the Qjump spelling checker and English, French and German dictionaries. If you want the massive dictionary that goes with *Perfect Plus* (one of the biggest computer dictionaries ever compiled), the price tag goes up to nearly £120.

The first Quill substitute to emerge was *The Editor* from digital Precision. As its title suggests, it was a text editor rather than a word processor, and had no dedicated means of producing anything sophisticated in the way of printed output. It was followed by the Special Edition of *The Editor*, which included a command to make a file into a document, and came with a separate printer driver.

# THREE

Man's best friend – the word processor. People become so devoted to theirs that they never change until circumstances force them. Others spend their lives in search of 'WP Right'. How do you choose, asks three-timer Mike Edwards? Is it the facilities – or just chemistry?

I took to *The Editor* like a duck to water. Unlike Quill, it doesn't use menus (which I dislike anyway) and instead offers a list of about one hundred commands which you enter and execute directly from the keyboard. The commands you normally need are easy to remember after a little practice, and give you comprehensive control over the file you are working on. There are simple, fast search and replace commands; it's easy to perform block operations and you can save command files to perform regular tasks. The Editor makes it fairly easy, for example, to take a list of exchanges and telephone codes, reverse it into a list of codes and corresponding places, and rearrange the list into numerical (or some other) order – all automatically, through a command file. You just couldn't do that with Quill. Nor can you do it with any other word processor that I know of and that is one good reason why I see no likelihood of giving up *The Editor*! There is a penalty for this flexibility: if you want to make full use of *The Editor*, you have to be prepared to think quite hard at times.

The problem with *The Editor* is the separate printer driver, which I couldn't get on with. The link between it and *The Editor* seemed tenuous, and using it was like being on an escalator – one way only. Instead, I wrote my own printer driver, with a nice friendly front end, rather like the *Turbo* front panel, but with more on it, so that you can change your mind about pretty well everything before actually issuing the fatal instruction to print. I would not be surprised if many other users have done the same, or have used *The Editor* for preparing text to be read in finally to Quill, or perhaps *text87*.

### Introducing text87

The next contender in the field was *text87*, which has now reached version 3. This is a proper word processor, with a built-in printer driver. It also makes strenuous efforts to be friendly towards Quill users in an effort to win them away from the original program. It uses menus, even though they are at the 'wrong' end of

```
Editing file: winl_ed_up_survey
File type : Text
Margins: L: 1 Ind: 1 R: 72
Tab interval: 3
Right just. : No Block type: Line
Block start : 0= Undefined
Block end : 0= Undefined
Marker point: 0= Undefined
Last search : F//
Repl. string: //
Buffer sizes: 87934 15368
Buffer usage: 48% 23%
```

- Press any key to continue -  
alternatives.

The field is so rich, that I'm going to restrict myself severely to a few of the many programs which could conceivably be used for word processing. I shall leave out text editors in which printout is a secondary consideration, so I shall ignore *Spy*, *Master Spy*, *QD* and *The Editor* in its ordinary version. (But I shall include *The Editor Special Edition* which has a proper printer driver.) I shall also avoid desktop-publishing programs which attempt to provide methods of laying out pages of text and graphics, and in which it is usually easier to read in text from a file created by some other program. This means

@ Line: 1 Col: 1 Line count: 598 Mode: Insert

**The Editor: the information screen and text style.**



# IN THE HAND

the page, and some of the items look similar to Quill. But there the similarity ends. Text87 is a much more sophisticated piece of software which offers as nearly complete control over both screen and printed output as it is possible to wish for. Results from it are super.

It is also a bit of a pain to work with, because however 'fast' it is (and it does what it does very quickly indeed), it is also slow, because of the complexity of its operations. The other major drawback of text87 is its manual. Very nicely presented, the manual has a big red logo in the middle of every page to distract the reader or discourage photocopying. It also fails to explain clearly a great deal.

## It's Perfection

The latest word processor is **Perfection** – again from Digital Precision, but very much more than a re-vamping of The Editor. Every effort has been made to make Perfection easily accessible to Quill users. There is the same system of menus (but more than in Quill) and in much the same format, in the same position on the screen. There is also for those who dislike trundling through menus, a system of keypresses to do the same jobs but faster. In lots of ways, Perfection is more intelligent than Quill and, above all, it is much quicker – very quick indeed in many of its operations. Searches are extremely fast. File-handling is as quick as anyone has a

right to expect. Perfection is quicker to load, and loads and saves files faster than its competitors.

So: how do you choose? Here is a point-by-point comparison which may help you. It's only fair to add that it doesn't help me to eliminate any of the programs: I shall still use all three, for different purposes.

Since I am going to spend quite a lot of time carping about more or less minor flaws in these programs, it would be as well to state here that all are excellent, well-documented pieces of work.

Manuals are a sore point with computer users. Their philosophy is: if all else fails, read the manual. Everyone wants to get on with the computing with as little trouble as possible – a fact which producers of hardware are well aware of! But you cannot expect to make full use of a word processor, or any other piece of sophisticated software, without taking some time and trouble over it.

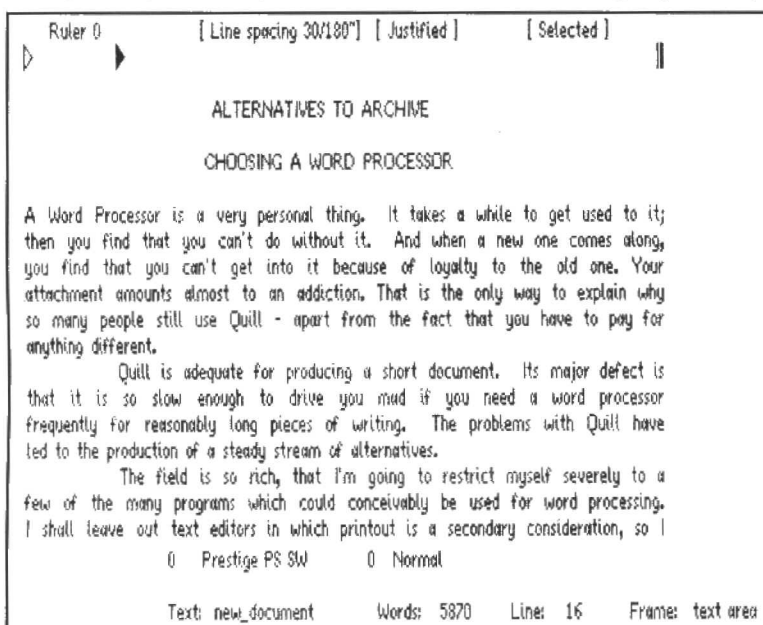
All three manuals are heavyweights. The Digital Precision manuals show a strong family resemblance. The **Perfection** manual has 104 A4 pages. This is a lot shorter than the manual for The Editor, which has 150 A4 pages. However, **The Editor's** 100 or so commands are summarised on just over two sides – and this is the bit of the manual that will need most attention, with only occasional reference to the rest to be sure exactly what a particular command does or how it is used. The commands are grouped in alphabeti-

cal order which happens usefully to be much the same as their grouping according to function. Much the same is true of Perfection: again, two or three sides contain the essential reference material – though the grouping here is less helpful, as it happens. There is an interesting disparity between the spirit of the DP advert, which refers to 'jargon-free, friendly but to-the-point documentation (that you will probably never need to read through!)', and the documentation itself, which more than once (rightly, in my view) advises study of the manual. Although the manual is very thorough on the whole, I have managed to generate just one error that I cannot find there (to be specific: 'picture-printer absent').

With **text87** you get a more standard type of handbook – A5, comb-bound, and very nicely printed, except that you can get very tired of the big red logo splashed across every page. Otherwise, I found this the least approachable manual. It is hard to find what you are looking for, even with the help of the index, which sometimes doesn't mention what you're hunting for ('proportional spacing' makes no appearance there, for example; nor does 'spacing'). The essential information about the command system occupies almost 30 pages – too long for quick reference, and although at times quite verbose, it's often not very helpful. An example is the explanation of the Feature command in the Context sub-menu, which, we are told, 'Selects the font to represent the whole group of Printtypes that are in the selected Feature'. I know text87 quite well, but I still don't quite understand that. What it is supposed to mean to a new user, I can't imagine. Writers of manuals need to keep in mind that it is more important to be comprehensible than strictly accurate.

## Know it first. . .

Of course, it has to be said that some of the things the program does are complicated, so you can't expect the manual to be simple. But I constantly feel that this is a manual with which, like so many manuals, it is best to know everything before you start reading it. On many topics, the index contains several page references – and all too often, none of them tells you what you need to know to use a facility. It took me hours to work out how to get page numbers in the middle of the page, (or anywhere), and I'm not, I believe, more stupid than the average QL user. The manual tends to hint in the general direction of an explanation, and then leave it up to you.



Text 87 showing the ruler and typeface



# WORD PROCESSORS

The problem could be resolved quite simply: What is required is a new section entitled 'How to prepare a document', which would run quickly (much more briefly than the present 30 pages on 'The Document', at any rate) though the basics, and indicate where to find more information.

Only the **Perfection** manual fails to offer an index – a major failing for any reference work in my view, but we understand that a full index will become available. It does however include a glossary which sometimes tells you where to look for details. It also includes material that has nothing to do with the program; for example, it explains the term 'proportional spacing' – which the program does not offer. This helps to put the functions in context.

All three programs have configuration systems, and they all allow you to change a great deal. There were no problems configuring **The Editor**, and the other two programs worked well in general. There are a few grouches about them, and some not so minor. In fact, I was unable to persuade either **Text87** or **Perfection** to do exactly what I wanted them to do without using **The Editor** to amend the task files directly.

## Configured out

This was a minor problem with **Text87**. You could say it was rather an effect of my choosiness about filenames than anything else. I wanted to use sub-directories on hard disk, and **text87** seems not to have been designed with that in mind. No doubt Version 4 will take hard disk on board.

Configuring turned out to be difficult with some aspects of **Perfection**. One problem is with the page which amends device and file names. Every time I got to 'Current default font device file name', the Configurator locked up – without my trying to alter anything! Since I cannot imagine DP not noticing this effect, it is likely to be something to do with my particular set-up.

My other, more general grouse about these programs (and about configuration programs in general) is that they are one-way systems: no going back or second thoughts allowed; you have to start again at the menu. A control panel (such as DP themselves use with **Turbo**) seems a more user-friendly way of doing it.

I would say from my own experience that **The Editor** is quicker to type into than **Perfection**. This is partly because **Perfection** is doing so much more. The other part is **Perfection's** personal way of doing things. When you are right at the beginning of a document, **Perfection** has a disconcerting habit of constantly re-writing the whole screen, or panning to left or right. Although you know that you can continue typing (the manual tells you so, and experience bears it out), it's nevertheless distracting. A real no-look touch-typist might get on a lot better than me. The **Editor** does the same sort of thing, but to

FEATURES		OPTIONS		SELECTIONS	
1	Prestige PS SW	1	Normal	Alpha	0 0 Prestige PS SW 0 Normal
2	Normal UL	2	Bold	Alpha	1 0 Prestige PS SW 0 Normal
3	Bold UL	3	Bold UL	Alpha	2 0 Prestige PS SW 0 Normal
4	SupS	4	SupS	Alpha	3 0 Prestige PS SW 0 Normal
5	SupS UL	5	SupS UL	Alpha	4 0 Prestige PS SW 0 Normal
6	SubS	6	SubS	Alpha	5 0 Prestige PS SW 0 Normal
7	SubS UL	7	SubS UL	Alpha	6 0 Prestige PS SW 0 Normal
8	SupS Bold	8	SupS Bold	Alpha	7 0 Prestige PS SW 0 Normal
9	SupS Bold UL	9	SupS Bold UL	Alpha	8 0 Prestige PS SW 0 Normal
10	SubS Bold				9 0 Prestige PS SW 0 Normal
11	SubS Bold UL				
12	Outline		Alpha		
13	Shadow		Alpha		
14	Shadow Outline		Alpha		

Select Feature Option Key View <esc>  
 [Select] a Printtype or its [Feature] or [Option] only; choose the [Key]; [View] the typefaces  
 Text: new\_document Words: 5870 Line: 16 Frame: text area

**Text87: the typeface selection screen**

a much less distracting extent. It also seems to work faster in re-orienting text on screen. The **Perfection** manual tells you that it's a good idea to set the right-hand margin sufficiently to the left to avoid constant panning. I like to work with 80, or at least 72 columns, both of which fit on the screen, but which involve panning until the text gets big enough to discourage the program from doing it. The **Editor** will cope with my settings happily.

**Text87** has a different philosophy. The characters are not standard QL sizes, so you can fit more or less as many as you want on to the screen, and vary the size of them at will; so no problem with panning. **Text 87** also reformats automatically as you type in, so that if you insert into existing text, it will all move along or down to accommodate what you have included. This can slow things down, producing a rather 'spongy' feel to your typing, and can also be distracting. In contrast, when you insert text into **The Editor** and **Perfection**, you often have to issue a command to re-format. Re-formatting can take time, even with a short paragraph.

## 100 commands

All three programs have around one hundred commands. As you would expect. **The Editor** offers many more options for doctoring the text, numbering or reordering the lines, and searching and exchanging. This is, after all, what it is designed for. The other two programs concentrate on preparing text for printing.

Both **The Editor** and **Perfection** have Help pages, and those in **Perfection** are particularly clear, and in alphabetical order which to some extent makes up for the lack of indexing.

**Text87** has no help screen, and it can be confusing to tell which command gives what you want, particularly since some of

the commands are also sub-commands under a different heading. Ruler, for instance, is both a fully-fledged command, and a sub-command in both **Block** and **Doc**. Nevertheless, *once you get used to it*, the command system is very logical, and no more complex than it needs to be.

## First letter logic

**The Editor** has logically grouped commands. All those for cursor navigation, for instance, begin with c: cs for the start of a line, ce for the end, cp for the start of the next paragraph. Block commands begin with b. This arrangement usually gives clear clues to what command you need to enter next. Confusion can still occur. Metathesis holds special dangers: for example, pr (reformat a paragraph) differs radically in effect from rp (repeat command).

The commands in **The Editor** are very snappy. You tap F3 followed by the code you want – normally one or two letters, but if you want to exchange a string the command can be quite long. As there is no menu system, you must either remember the command, keep the manual nearby, or screen the help pages.

**Perfection** boasts both a menu system of four menus, and an immediate command system. F3 is used to cycle through the menus, and a single keypress then accesses any command on the menu. The immediate command system is a natural one, involving, as far as possible, the use of initial letters with combinations of ALT, CTRL and SHIFT. The system has been thought out carefully to avoid clashing with other uses (CTRL/c to cycle jobs, ALT/letter under **Qram** or **Qpac**). There are some minor annoyances, inevitably. If you use the menu system to toggle justification, you have to go through menu 0 three times to reach left-justification. Total key-presses: nine! But in the immediate commands, three times CTRL/



a does the same job. But why return to menu 0 every time?

Now for an example of how the command systems work in practice: we will replace 'The Editor' with 'Perfection' throughout a document, using each program in turn.

The command format for **The Editor** is:

F3 rp;e.The Editor.Perfection. ENTER

This would replace every occurrence of 'The Editor' with 'Perfection' throughout a document from the cursor point. F3 accesses the command system; rp tells the computer to repeat the operation; e means exchange the following strings.

To achieve the same effect in **Perfection**, you must first search for a string (CTRL/f). Having found it, you then enter a replace string (CTRL/v), and then give the command for an immediate replace CTRL/x) either backwards or forwards, depending on which direction is pre-set. Example:

CTRL/f The Editor ENTER CTRL/v Perfection ENTER CTRL/x

You can do the same thing more slowly through the menu system.

Unlike the DP programs, **text87** offers only case-sensitive search, and would entirely fail to find 'text87' in this article. (Perhaps I ought to point out here that Software87 themselves use a small initial 't'.) Like all the other commands, you have to use the menu system, typing F3 first to access the menu, then s for Search. Finding your string is fast and easy once you've got the case right, and to exchange you need to include a special marker character obtained with F5. The example is:

F3 s The Editor F5 Perfection ENTER A

The final A is required because, when the program finds the search-string, you are asked if you want to Replace, Continue searching, replace All further occurrences, or ESCape.

Isn't it amazing that there are three such different methods of tackling the same task?

All three programs offer comprehensive file handling facilities. They include merging files, saving native files, exporting files, importing foreign files and saving selected parts of a file.

The programs offer different methods of reading 'foreign' files. **The Editor** will read anything — ru is the command for Quill or other odd files, while plain r does for Editor files. The unformatted read allows you to work on binary files, executable jobs, or whatever. Just as well: I would never have been able to customise Text87 or Perfection to my personal requirements without this facility!

### Importing text

**Text87** offers you the chance to import Ascii, Quill or text87 Export files. In the case of an Ascii file, you are invited to read in 'normal' (ie as it comes) or 'paragraph' format (ie with paragraphs formatted as in Text87).

**Perfection** is easily the cleverest program in this respect. It will read, recognise, and act appropriately on Perfection, Quill, or plain text files without any user encouragement. Great! However, if you want to export a plain text file, you must either use a separate supplied program, *Stripsort*, to remove all the bits and pieces that Perfection puts into the text, or else set up a printer driver to do the same job. I should add that the primary aim of Stripsort is, as its name implies, to sort material as might

be required if Perfection is used for database handling: the stripping of attributes from a file is secondary.

Unlike Quill, all these programs read complete files into memory. This means that memory limitations can be a problem. Most programs expect you to set memory limits before reading in a file. **The Editor** works like this. In the case of **text87**, you have to set the limits before executing the program itself. If insufficient space is allocated, you have to reset memory limits, and re-execute the program before the new limits can become operative. Again, Perfection is very smart and organises its own memory dynamically to reflect the size of the file you want to work with, avoiding all these problems.

The DP programs both warn you if a file already exists, and ask if you want to overwrite. Only text87 operates a sudden-death system of saving exactly what you tell it to, without any queries. So text87 is convenient, but only if you're 100% sure you know what you're doing! How many of us can say that at all times?

The display is very clean in all three programs. Screen size is user-configurable. All three programs offer, or can be configured to offer, just a blank rectangle and a status panel at the bottom of the screen. All three programs incorporate an automatic word count, even The Editor if you ask it to treat a file as a document. But then even Quill did this. **Perfection** can have, and defaults to showing, a menu at the top of the screen. **Text87** keeps three lines at the bottom of the screen for menus, status information, and user responses. You can also show file information (the current ruler, in fact) at the top of the screen. **The Editor** uses just one line at the bottom of the screen for status information, command entries and reports. Each program can display about 15-24 lines of text at a time.

**The Editor** is simplest of the three programs. There are two alternative screens: the Help screen, and the status screen, which usefully shows you which defaults you are operating with — left and right margin positions, the current file-name, and so on.

### Status lines

In **Text87** the status line shows the current line number and the number of words in the document. In some modes, it shows a variety of other information. The command line is blank until you switch on the relevant menus with F3 and sub-commands. Another line is available for entering responses, or for further information. There the simplicity ends, however. Text87 has an array of other screens to deal with. There is a page-layout screen to help you set the overall parameters of your document. Then there is a Ruler screen which helps you to set tabs and margins and line spacing. Finally, there is a Type screen

## ALTERNATIVES TO ARCHIVE

### CHOOSING A WORD PROCESSOR

A Word Processor is a very personal thing. It takes a while to get used to it; then you find that you can't do without it. And when a new one comes along, you find that you can't get into it because of loyalty to the old one. Your attachment amounts almost to an addiction. That is the only way to explain why so many people still use Quill — apart from the fact that you have to pay for anything different.

Quill is adequate for producing a short document, but it has difficulty handling long documents. Its range of operations is restricted. Its major defect is that it is slow enough to drive you mad if you need a word processor frequently for reasonably long pieces of writing. The problems with Quill have led to the production of a steady stream of alternatives.

The field is so rich, that I'm going to restrict myself severely to a few of the many programs which could conceivably be used for word processing. I shall leave out text editors in which printout is a secondary consideration, so I shall ignore Spy, Master Spy, QD and The Editor in its ordinary version. I shall

Text: new\_document Words: 6283 Line: 1 Frame: text area

Text87: display of different character sizes



# WORD PROCESSORS

which you can use to select which typefaces your printer will use.

In text entry mode, text87 has a useful toggle which allows you to show the attributes of the current line. Its ruler (margins and tabs) appear at the top of the screen, while the status line at the bottom shows the typeface used.

Text87 lets you alter the size of characters on the screen. You can select different

display the menu on start-up. You can, of course, toggle the menu display on or off with F2. Otherwise, there is just a plain screen with a status line at the bottom. On start-up, the page appears to occupy a fixed number of lines regardless of whether or not the program is configured to display the menu. If you configure not to use the menus, the typing rectangle normally has some blank lines at the bottom correspond-

One of the most useful facilities in **Perfection** is that you can operate two document windows at the same time. You can quickly toggle between them to work on, say, the beginning and the end of your blockbuster at once. Only Perfection offers automatic page-diversion lines on screen.

## Page divisions

**The Editor** does so only if you ask it to make a document of a file. **Text87** shows page-divisions only if you ask it to Goto a page - though once you have done that, you can happily scroll through from one page to the start of the next. It is quite easy, too, to find yourself dealing with some very odd page numbers; and it is quite hard to work out how to get them back to normal. There are, I feel, improvements that could be made in this area of the operation of Text87.

**WYSIWYG** (what you see is what you get) is prized by writers laying out documents. **The Editor** doesn't have it. Whatever control codes you put in will stay there on the screen, taking up space and distorting the line lengths of whatever material you have typed in.

## WYSIWYG

**Text87** has WYSIWYG. Double height characters look it, and so do quad size. Compressed print looks squashed up, though not by much. Proportional print is presented proportionally on the screen. Italic usually doesn't look italic in my setup, but could be made to do so. Bold is a different colour. Different printer fonts are not distinguished by default, but there is a command to Attach screen display fonts to different printer effects, and you can vary the number of screen display fonts you use. Anyone with a gold card might be tempted to greater extravagance with memory.

**Above:**  
**Perfection,**  
the split  
screen.

**Left:**  
**Perfection**  
with one of  
the Menus  
displayed

### ALTERNATIVES TO ARCHIVE

#### CHOOSING A WORD PROCESSOR

A Word Processor is a very personal thing. It takes a while to get used to it; then you find that you can't do without it. And when a new one comes along, you find that you can't get into it because of loyalty to the old one. Your attachment amounts almost to an addiction. That is the only way to explain why so many people still use Quill - apart from the fact that you have to pay for anything different.

Quill is adequate for producing a short document. Its major defect is that it is so slow enough to drive you mad if you need a word processor frequently for reasonably long pieces of writing. The

	QUILL	EDITOR	TEXT87	PERFECTION
EXECUTE PROGRAM	10	5	9	4
LOAD FILE	9	7	4	4
IMPORT FILE	51	7	10	3
SAVE FILE	7	7	3	2
GOTO BOTTOM/TOP	6	1.5	2	1.5
MOVE BLOCK	155 (!)	18	17	25
REPLACE STRING	48	13	12	25
COL1 LIN491 /599 PAGE	LJ	VL	#10	

scales and compression ratios to correspond to what is on the printed page. I use the default scale, 60%, which easily permits 80 columns and more on the screen.

**Perfection** lets you choose whether to

won't normally go away until you press F4 to re-draw the screen, the Help pages. Toggling the menus on an off seems to remove this effect, and you can then play with the whole document window.

F1 HELP                      SHIFT/F1 last cmd    ESC Escape/Cancel    ↑↓ by char/line  
F2 Menu on screen    SHIFT/F2 Mode8 ++    SHIFT/F5 one/two - + SHIFT para/word  
F3 Menu 1+2+3+1    SHIFT/F3 Menu3+2+1 windows, F5 selects + SHIFT/ALT window  
F4 Refresh screen    SHIFT/F4 Size/Move ALT/F5 Over/Insert CTRL/C toggle job

### ALTERNATIVES TO ARCHIVE

#### CHOOSING A WORD PROCESSOR

A Word Processor is a very personal thing. It takes a while to get used to it; then you find that you can't do without it. And when a new one comes along, you find that you can't get into it because of loyalty to the old one. Your attachment amounts almost to an addiction. That is the only way to explain why so many people still use Quill - apart from the fact that you have to pay for anything different.

Quill is adequate for producing a short document. Its major defect is that it is so slow enough to drive you mad if you need a word processor frequently for reasonably long pieces of writing. The problems with Quill have led to the production of a steady stream of alternatives.

The field is so rich, that I'm going to restrict myself severely to a few of the many programs which could conceivably be used for word processing. I shall leave out text editors in which printout is a secondary consideration, so I shall ignore Spy, Master Spy, QD and The

COL1 LIN1 /0 PAGE                      LJ                      VL                      #32



**Perfection** is a kind of half-way house. Bold (or emphasised) print is a different colour on screen, and there are three different strips which can be used for different printed effects.

Different sizes of print aren't shown exactly as they will appear, nor are different printer fonts. Italics, unusually, do appear so, and there are also fonts for subscript and superscript which are very realistic. For my money, **text87** is the WYSIWIGiest of the three programs, if WYSIWIGness is what you want. I think it is over-rated. Both **Perfection** and **Text87** have clear ways of distinguishing a range of special effects.

## Printer drivers

**The Editor** is distinct in having a separate printer driver. I could not be happy with it: it functions too much like a one-way escalator, and will not permit changes of mind, but it does, of course, work well as a printer driver.

The **Perfection** driver functions from within the word processor, and has a brief interactive exchange before setting off. The driver is very complete, with up to sixteen translates which can be used for any printer features, such as different fonts, double strike, or whatever, in addition to a good range of standard features. Should you find that the whole battery of effects is insufficient, you can create several printer drivers, and load in whichever is most appropriate for specific needs. It would be possible to keep one for drawing neat IBM-style boxes, for instance – but you can only use one driver per printing session. Printing out something simple, such as a letter, has been made very quick and easy.

## Art of creation. . .

As usual **text87** adopts a different philosophy. You can create your own printer driver, but it's not easy. **Software87** obviously don't expect many users to do so, and instead provide an impressive battery of complete printer drivers. I use a Star XB-24, and have a range of about 200 typefaces to choose from. That seems a lot, but every variation, no matter how minor, is a different typeface – Courier 10 cpi is different from Courier Proportional, for instance. Annoyingly, I can select only 10 of these at a time, and I have to re-select them every time I want to use the program: there is no facility for saving this selection other than as parts of an existing document, and I feel there ought to be. Perhaps this facility will be incorporated in a version 4 **text87**. Until then, you can get by with saving dummy documents incorporating the required typefaces.

Although **Perfection** can be programmed, via the printer driver, to use a proportional font, it cannot produce a proportional

screen, nor can it automatically produce a fully justified proportional printout. The program always counts characters, not pixels. The manual suggests (p.23) that **DP** may get round to producing a module for proportional output eventually – a hint of **Perfection Super-Plus**? Until that time, **Text87** is the only route to a really handsome proportional output with full justification.

## Home speed trials

The speed at which a program works depends on extraneous factors like what is resident in the computer, its hardware configuration, how well it feels, and how well the user feels, as well as on intrinsic considerations. My timings are by stopwatch, in seconds, and very rough. These are not conventional benchmarks. Where a program takes varying times to perform a task, I have given here an average. Where file operations are involved, the timings are for hard disk:

	Quill	Editor	text87	Perfection
Execute program	10	5	9	4
Load file	9	7	4	4
Import file	51	7	10	3
Save file	7	7	3	2
GoTo bottom/top	6	1.5	2	1.5 (long files, too)
Move block	155(!)	18	17	25
Replace string	48	13	12	25
Reformat paragraph	automatic	7	automatic	15

All the timings are subjective. It would be fair to say that I am less familiar with **Perfection** than with the other programs, but I did try to live up to the advertising and do things fast. User input is most important in the block move operation, and the repeated string-replace operation. Such operations are, in my view, easier in **The Editor** than in either of the other programs.

## Your excellency

Each of these programs is excellent in many respects; none is perfect. I feel that **The Editor** offers the most trouble-free editing of text – which is what it is intended for. However, I dislike its printer driver, and instead use one of my own devising. **Text87** offers a display which is pretty much as close as you can get to WYSIWYG, and has the advantage of allowing true proportional printout. The program allows a remarkable degree of user configuration. It also has complex printer drivers inbuilt, and includes the QJump spelling checker. Its disadvantages are lack of automatic page-division indicators, and the amazing manual. The screen which allows you to view the typefaces is not useful – it would take ten years to scroll through all those available,

I guess – and the inability to save and reload a selection of typefaces is a drawback. The range of text-handling facilities is smaller than in the other programs – there is only one kind of search/replace, for instance.

**Perfection** has, most of all, simplicity of operation, and a general familiarity for Quill users. There is also the very valuable split screen, the convenient printer driver, user-configurability, the speed of many of its operations, the alternative of menu-driven or immediate commands, the good range of commands, and the intelligent file-handling operations. **Perfection** is a pretty assertive title, however, and as Mr Knightley says of Emma at Box Hill, "Perfection should not have come so soon." The program is still a bit raw in its version 2 state. The block-handling is – in my erratic hands anyway – unreliable; I think there must be easier ways to define a block. However, version 3 is on the market as we speak and most of the glitches (bugs is too strong a word) have been ironed out.

It is not easy to choose between these programs, so I shall continue to use all of them. For articles like this, **The Editor** is indispensable for the ease with which blocks of text can be moved from place to place, and for its fast reaction time. For producing semi-DTP-type material, I see no sensible alternative to **Text87**. But this is a program for people who are prepared to devote a lot of time to becoming familiar with its facilities. **Perfection** will be ideal for letters and documents. It will appeal particularly to those who want speed and simplicity of operation above all else.

*So what do you do? Well, it's been said that changing your word processor is like ending a love affair. You wave goodbye to the time and emotion you invested – and then you have to start all over again. Mind you, it can be like falling in love again. And do you choose simplicity, or the all-singing, all-dancing sophistication? They say 'wealth won't bring happiness, but it helps you to be miserable in comfort'. The same goes for a wealth of facilities. What about Quill users and Perfection? If you like the mother, you'll probably like the daughter – even if she seems a bit brainy to begin with. Trust your instincts – it's more fun, too.*



# DIY TOOLKIT

Simon Goodwin presents fast QL code to compare decimals and search arrays.

```
* QL WORLD DIY TOOLKIT - FAST COMPARISON FUNCTIONS
* Version 0.7, Copyright 1992 Simon N Goodwin.
*
initialise lea.l    define,a1
move.w    $110.W,a2    BP.INIT vector
jmp      (a2)

*
* MININT returns the lowest valid integer of an array or list
*
minint     cmpa.l    a3,a5    Any parameters?
beq.s     lowest    If not, return -32768
hrr.s     get_ints
compare   cmp.w     ~0(a2,a6.1),d0    Is D0 less than (A2,A6).?
ble.s     too_much
move.w    0(a2,a6.1),d0    No, pick up the new lowest
addq.l    #2,a2
dbra      d3,compare
stack_int move.w    d0,0(a1,a6.1)    Stack the highest found
got_int   moveq     #3,d4    INT result is stacked
moveq     #0,d0    No error
rts

*
* Return constants -32768 or +32767 if there are no parameters
*
lowest     move.w    ##8000,d4    MINIMUMX is -32768
bra.s     return_int
lots       move.w    ##7FFF,d4    MAXIMUMX is 32767
return_int moveq     #2,d1    No. of bytes needed
*
* Check there are D1 bytes free for the result on the RI Stack
*
stack_chk  move.l    d1,d7    Remember the space needed
movea.w   $11A.W,a0    Read the BV.CHRIX vector
jsr      (a0)    Check for D1 stack bytes
stack_num  movea.l   $58(a6),a1    Get a safe A1 value
suba.l    d7,a1    Allocate the space
move.l    a1,$58(a6)    Update BV.RIP
move.w    d4,0(a1,a6.1)    Stack one word
subq.l    #2,d7    Is that enough?
beq.s     got_int
move.l    d5,2(a1,a6.1)    Stack mantissa
bra      got_float

*
* MAXINT finds the highest valid integer it can
*
maxint     cmpa.l    a3,a5    Any parameters?
beq.s     lots       If not, return +32767
bsr.s     get_ints    Put D3+1 params @ (A6,A2)
compare4   cmp.w     0(a2,a6.1),d0    Test one
bge.s     not_top4
move.w    0(a2,a6.1),d0    Pick up the new highest
addq.l    #2,a2    Advance to the next
dbra      d3,compare4
bra.s     stack_int    Return D0 to SuperBASIC

*
get_ints   move.l    $58(a6),d7    Save BV.RIP for later
movea.l   (a7)+,a4    Tidy up return address
moveq     #2,d5    Number of bytes per value
cmp.w     ##0303,0(a3,a6.1)    Integer array?
bne.s     misc_ints    No, read integer parameters
lea.l     get_ready,a5    Continue there later

*
* FIND_ARRAY start with (A3,A6) -> NT Entry, D5 = Element size
* Exit @A5 with first word in D0, (A2,A6) -> the 2nd element
*
find_array movea.l   4(a3,a6.1),a3    A3 -> Array descriptor
movea.l   40(a6),a2    Fetch BV.VVBAS offset
adda.l    a2,a3    A3 is descriptor A6 offset
cmp.w     #1,4(a3,a6.1)    Check number of dimensions
bne.s     bad_param    Require just 1 dimension
move.l    d5,d1    Ensure RI space for result
movea.w   $11A.W,a0    Fetch BV.CHRIX vector
jsr      (a0)    No return if this fails
move.w    6(a3,a6.1),d3    Fetch maximum index value
subq.w    #1,d3    Prepare for DBRA later
bmi.s     bad_param    Require 2+ array elements!
adda.l    0(a3,a6.1),a2    A2 is A6 offset of values
move.w    0(a2,a6.1),d0    Pick up the first word
adda.l    d5,a2    Advance to the next element
jmp      (a5)

*
```

This month's listings introduce four simple functions and two new ideas: array parameters and fast floating-point operations. Array parameters allow tens or thousands of values to be processed by one resident procedure or function. Much more speed is attainable than when processing array elements one at a time in Basic.

I also explain the organisation of Qdos floating point numbers, showing how knowledge of the format allows faster comparisons than are possible with rom routines.

The new functions are called MAXIMUM and MINIMUM, with speedy integer equivalents MAXIMUM% and MINIMUM%. The effect depends on the number of parameters you supply. Called without parameters, the functions return the largest or smallest number that can be represented, so MAXIMUM% returns 32767, the integer limit, while the floating-point MAXIMUM returns 1.61585 e616. Alternatively, you can pass the name of an array, and the function will scan all the values inside at great speed, returning the highest or lowest as appropriate.

The most obvious case is when you pass a sequence of values, variables or expressions. The functions evaluate all the parameters and return the value of the highest or lowest. You may pass any number of parameters for comparison.

MAXIMUM% (0,x%) returns X%, or zero if X% is less than zero. Such a call might be used to constrain pixel co-ordinates, avoiding an 'out of range' report. MINIMUM (a,PI) returns either the value of A or 3.141593, whichever is the lowest. Without parameters MINIMUM% returns -32768, the lowest integer value.

The functions are very fast when scanning large arrays, even in interpreted Basic. The integer functions scan over 84,000 values per second on my expanded QL, and floating-point comparisons run at 24,500 per second. The routines in **Listing one** locate arrays for many other purposes besides finding maxima and minima.

When used with no parameters, or a list of expressions, these functions should suit all known roms, compilers and emulators. There is no limit on the number of parameters which may be supplied.

The facility to pass an array parameter relies on the interpreter's Name Table and array descriptor system, documented in 1985 by Jan Jones in *QL SuperBasic, The Definitive Handbook*. Array parameters work fine with Sinclair roms, Minerva, Thors and *QLiberator* tasks.

*Supercharge* and *Turbo* work differently; the equivalent of the Name Table is half the size, and descriptors are rather more efficiently organised. Alas, array parameters are not supported, and there is no easy way to locate descriptors inside a task; you get a compile-time error report if you try to pass a compiled array to a resident function.

Listing 1 is the source code for the new functions. You can type this into your own assembler if you wish to re-assemble or modify the code, but you do not need an assembler to use the code as written.

**Listing two** generates the code file directly, reading the instructions from SuperBasic DATA statements. Lines 100 to 580 of the loader are common to each DIY



```

misc_ints  movea.w  $112.w,a2      Vector to get integers
          jsr      (a2)             CA.6TINT
          bne.s    bad_exit
          subq.w   #2,d3           Only 1 parameter?
          bmi      got_int         Easy, it's already stacked
          move.w   0(a1,a6.1),d0   Pick up the first integer
          move.l   a1,a2
          adda.l   d5,a2           Point A2 at the next
get_ready  movea.l   d7,a1         Retrieve old BV.RIP
          suba.l   d5,a1         Make room for the result
          move.l   a1,$58(a6)     Set BV.RIP for later
          jmp      (a4)           Return wherever

*
bad_param  moveq     #-15,d0       BAD PARAMETER error
bad_exit   rts                  Return error code in D0
*
get_floats move.l   $58(a6),d7     Save BV.RIP for later
          movea.l   (a7)+,a4       Return point if all's well
          moveq     #6,d5          Allow six bytes per element
          cmp.w     #0x02,0(a3,a6.1) Floating point array?
          bne.s    misc_float
          lea.l     get_mantis,a5  Continue from here later
          bra.s     find_array

*
misc_float movea.w  $114.w,a2      Vector to get floats
          jsr      (a2)             CA.6TFP
          bne.s    bad_exit
          subq.w   #2,d3           Only 1 parameter?
          bmi.s    got_float
          move.w   0(a1,a6.1),d0   Pick up exponent
          lea.l    6(a1),a2        Point A2 at the next
get_mantis move.l   -4(a2,a6.1),d1 Don't forget the mantissa
          bra.s    get_ready

*
* Find MINIMUM floating-point value; constant, array or list
*
minimum    cmpa.l   a3,a5          Any parameters?
          bne.s    get_minis
          move.w    #0xFFF,d4      Exponent is 4095 decimal
          move.l    #0x000000,d5   Mantissa has 7 hex zeros
          bra       stack_chk      MINIMUM is -1.61585 e616

*
get_minis  bsr.s    get_floats     Find the parameter(s)
compare3   bsr.s    compare_fp     Is D0 & D1 the lowest ?
          bcc.s    not_least
          move.w    0(a2,a6.1),d0   Pick up the new lowest
          move.l    2(a2,a6.1),d1   exponent and mantissa
          addq.l    #6,a2           Advance to the next one
          dbra     d3,compare3     Try up to 65536 times
          bra.s    stack_fp        Put D0 and D1 on RI Stack

*
* Find MAXIMUM floating-point value; constant, array or list
*
maximum    cmpa.l   a3,a5          Any parameters?
          bne.s    get_maxes
          move.w    #0xFFF,d4      Exponent is 4095 decimal
          move.l    #0xFFFFFFF,d5  Mantissa has 7 hex 'F's
          bra       stack_chk      MAXIMUM is +1.61585 e616

*
get_maxes  bsr.s    get_floats     Put D3+1 values @ (A2,A6)
compare2   bsr.s    compare_fp
          bls.s    not_top2
          move.w    0(a2,a6.1),d0   Skip if Z or C flag set
          move.l    2(a2,a6.1),d1   Pick up the new highest
          addq.l    #6,a2           Don't forget the mantissa
          dbra     d3,compare2     Try up to 65536 times

*
stack_fp   move.w    d0,0(a1,a6.1) Stack the exponent
          move.l    d1,2(a1,a6.1) Stack the mantissa
          moveq     #2,d4           Flag FLOAT result stacked
          moveq     #0,d0          No error
          rts

*
* FAST F.P. COMPARE, sets C & Z for 0.5(A2,A6) - D0.W D1.L
*
compare_fp move.l    2(a2,a6.1),d5   Pick up next mantissa
          move.l    d1,d4           Copy the highest yet
          eor.l     d5,d4           Do the signs match?
          bpl.s    same_sign
          tst.l     d1             No; is best yet -ve ?
          bmi.s    compared
          add.l     d4,d4           Set Carry, since D4<0
          rts

*
same_sign  move.w    0(a2,a6.1),d2   Pick up next exponent
          tst.l     d5             Are mantissae positive?
          bmi.s    both_minus
          cmp.w     d0,d2           If not, bigger means less
          bne.s    compared
          cmp.l     d1,d5           Compare exponents
          rts                     Easy if they differ
                                   Else check mantissae

*
both_minus cmp.w     d2,d0           Compare exponents
          bne.s    compared
          cmp.l     d1,d5           Exponents give result
          rts                     Mantissae give result

*
define     dc.w     0,0             No procedures
          dc.w     5               Four long-named functions
          dc.b     7,'MAXIMUM'
          dc.b     7,'MINIMUM'
          dc.w     maxint-*,
          dc.b     8,'MAXIMUMZ'
          dc.w     minint-*,
          dc.b     8,'MINIMUMZ'
          dc.w     0
          end

```

Toolkit project. Lines from 590 onwards are specific to this project. I have experimented with a faster, more friendly decimal loader, but shall stick with hexadecimal for the time being, after mixed reactions from readers.

Note that you can change the names after the code file has been generated as well, using a binary file editor such as *The Editor*, *Spy* or *Micro-Emacs*. Just load the file, select 'overtyping' mode, and type the new name in capitals over the original before saving the new file. It is important not to change the length of the file or it will not work thereafter.

The ALIAS command from DIY Toolkit Volume A, allows you to give extra names to any toolkit command or function. BPOKES\_FORGET\_BAS, in Volume B, lets you discard resident names if they clash with names in your existing SuperBasic.

I shall continue to use the hex loader to generate stand-alone code, but I shall use faster decimal for composite programs, like *Task Commander*, that run some Basic and read code from DATA when used. Assuming you specify MINMAX\_CODE as output from the hex loader, these instructions load the MAXIMUM and MINIMUM functions and link them to SuperBasic:

```

x=RESPR(448)
LBYTES mdv1_MINMAX_CODE,x
CALL x

```

The code and assembler source are available on disc or cartridge from **DIY Toolkit at Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA**. Send £7 for DIY Toolkit Volume 2, enclosing a microdrive cartridge if you do not have a disc drive.

Listing 1 includes full source code for the new functions. Much code is shared between all four; for instance, they all return via STK\_CHECK if called with no parameters, and they all use FIND\_ARRAY if passed an array. GET\_FLOATS fetches floating-point values for MAXIMUM and MINIMUM, while GET\_INTS does the same for the integer functions. These routines are shared in an unusual way which the 68000 supports, but is more characteristic of code for modern risc processors or 1950s computers without a 'stack'.

GET\_INTS and GET\_FLOATS are called with conventional BSR instructions, the equivalent of Basic's GO SUB, but both start by popping the return address into A4. This means the shared code can return errors directly to SuperBasic as if there was no intermediate call, performing RTS with the report code in D0. If all goes well the routines return with JMP (A4), and there is no need to check the error code.

A similar but faster technique is used to invoke FIND\_ARRAY, which may also need to return an error code. The program points A5 at the target address, using a position-independent LEA instruction just before FIND\_ARRAY is reached.

This is faster because there is no need to transfer the four byte return address to and from the A7 stack. The address register is used instead of the stack memory. Such techniques gobble one address register every time you use them, but they are interesting and can be useful.

Fast QL code minimises the number of bytes transferred to and from ram. The unexpanded QL is limited to about 860,000 byte transfers per second; this doubles

```

QL World DIY Toolkit April 1992, Listing 2:
100 REMark Sinclair QL World HEX LOADER v 3
110 REMark by Marcus Jeffery & Simon N Goodwin
150 CLS: RESTORE : READ space: start=ALCHP(space)
160 PRINT "Loading Hex..." : HEX_LOAD start
170 INPUT "Save to file...";f$
180 SBYTES f$,start,byte : STOP
190 :
200 DEFine FuNction DECIMAL(x)
210 RETURN CODE(h$(x))-48-7*(h$(x)>"9")
220 END DEFine DECIMAL
230 :
240 DEFine PROCedure HEX_LOAD(start)
290 byte = 0 : checksum = 0
300 REPEAT load_hex_digits
310   READ h$
320   IF h$="*" : EXIT load_hex_digits
330   IF LEN(h$) MOD 2
340     PRINT"Odd number of hex digits-in: ";h$
350     STOP
360   END IF
370   FOR b = 1 TO LEN(h$) STEP 2
380     hb = DECIMAL(b) : lb = DECIMAL(b+1)
390     IF hb<0 OR hb>15 OR lb<0 OR lb>15
400       PRINT"Illegal hex digit in: ";h$ : STOP
420     END IF
430     POKE start+byte,16*hb+lb
440     checksum = checksum + 16*hb + lb
450     byte = byte + 1
460   END FOR b
470 END REPEAT load_hex_digits
480 READ check
490 IF check <> checksum
500   PRINT"Checksum incorrect. Recheck data.":STOP
520 END IF
530 PRINT"Checksum correct, data entered at: ";start
560 END DEFine HEX_LOAD
570 :
580 REMark Space requirements for the machine code
590 DATA 448
610 REMark Machine code data
620 DATA "43FA017E34780110","4ED2BBCB671C6162"
630 DATA "B072E8006F043032","E800548A51CBFFF2"
640 DATA "3380E80078037000","4E75383C80006004"
650 DATA "383C7FFF72022E01","3078011A4E90226E"
660 DATA "005893C72D490058","3384E800558767D4"
670 DATA "2385E80278027000","4E75BBCB67D26112"
680 DATA "B072E8006C043032","E800548A51CBFFF2"
690 DATA "60AE2E2E0058285F","0C730303E800662E"
700 DATA "2873E804246E0028","D7CA0C730001E804"
710 DATA "663C72023078011A","4E903633E8065543"
720 DATA "6B2CD5F3E8003032","E800548A60163478"
730 DATA "01124E92661A5543","6B00FF6A3031E800"
740 DATA "45E9000222475589","2D4900584ED470F1"
750 DATA "4E75BBCB660E383C","0FFF2A3C80000000"
760 DATA "6000FF546126616E","64083032E8002232"
770 DATA "E8025C8A51CBFFF0","604EBBCB6636383C"
780 DATA "0FFF2A3C7FFFFFFF","6000FF2C2E2E0058"
790 DATA "285F347801144E92","66B655436B323031"
800 DATA "E8002231E80245E9","000622475D892D49"
810 DATA "00584ED461D6611E","63083032E8002232"
820 DATA "E8025C8A51CBFFF0","3380E8002381E802"
830 DATA "780270004E752A32","E8022801BB846A08"
840 DATA "4A816B12D8844E75","3432E8004A856B08"
850 DATA "B4406602BA814E75","B04266FABA814E75"
860 DATA "000000000005FF74","074D4158494D554D"
870 DATA "FF42074D494E494D","554DFEC0084D4158"
880 DATA "494D554D2500FE64","084D494E494D554D"
890 DATA "25000000","*',"40303

```

with rom or fast ram expansion.

You may be surprised to see the value 5, rather than 4, before the table of function names and addresses. The average length of names is more than the default maximum of seven, so an extra eight bytes of Name List space may be needed; you tell BP.INIT about this by adding one to the function count. Each unit expands the Name Table and Name List by eight bytes.

When a SuperBasic array is dimensioned, the interpreter allocates memory for the elements in the Variable Values area. It also creates a 'descriptor' which describes the shape and location of the array in memory. The routines check the Name Table entry for the first parameter. If it is an array the Table holds \$0303 for integers, or \$0302 for floating-point values. The dollar sign indicates the hexadecimal notation, base 16, which was used to design the Name Table. The third possibility is \$0301, for a string array.

The second half of the Name Table entry holds a long word, the offset of the array descriptor in the Variable Values area. SuperBasic locates each array element by looking up the array name in the Name Table, then finding the descriptor, which contains information needed to locate the elements themselves.

The size of an array descriptor depends on the number of dimensions. The minimum is ten bytes, for a one-dimensional array, and rises by four bytes for each additional dimension. The first long word of each descriptor points to the elements. It holds an offset relative to the start of the Variable Values area. All other values are 16 bit integers.

The next word holds the number of dimensions, followed by a pair of words for each dimension. The first word in each pair indicates the maximum subscript value for the dimension, and the second holds the distance between elements of this dimension, in units of one element: six bytes for decimals and two for integers.

DIM S\$(10) generates the offset, followed by 1, the number of dimensions, 10, the maximum subscript and 1, using one byte per element. DIM A(10,10) follows the offset with 2, 10, 11 (remember element zero!), 10 and 1.

DIM X(1,1,1) gives a three dimensional array, with descriptor words 3, 1, 4, 1, 2, 1, 1. The interpreter treats element zero of a string array as the two-byte string length, so DIM S\$(10, 10) stores descriptor values 2, 10, 12, 10 and 1 after the offset of the elements.

Array elements are stored sequentially, with the last subscript varying most quickly, so that DIM X(1,1) allocates space for X(0,0), X(0,1), X(1,0) and X(1,1), in that order.

As presented, the MINIMUM and MAXIMUM routines expect lists of values or one-dimensional arrays, but the code could be extended to process any shape of array, even a selected slice. The DBRA must be replaced with a long count in D3 if you need more than 65537 elements.

The CA.GT parameter routines lines up values on the RI Stack just like array elements. The same routines can scan either, once A2 holds the offset of the value. All the functions start by picking up the first value, and compare it with each of the others in turn. If you only supply one



parameter it is returned at once.

The functions report Bad Parameter (ERR\_BP) if you pass them too many dimensions, the wrong type of array, or an 'array' with only one element. The type of the array should match the type of the function, be it integer or floating-point.

The 32 bit registers in the QL can store whole numbers in the range -214748368 to 214748367. The 68008 and 68000 need combinations of instructions to manipulate numbers outside that range. The QL rom floating-point arithmetic can handle numbers in the range +/- 1.61585 e616.

The 'e' indicates that the following number is a count of decimal places, or exponent; the QL exponent allows a mind-boggling 616 digit places before and after the decimal point, but the 32 bit mantissa is only accurate to about nine decimal digits.

Exponent and mantissa are stored in binary. The exponent is a word between 0 and 4095, and indicates the position of a binary point in the mantissa. To double the overall value, add one to the exponent. To halve it, subtract one from the exponent.

INPUT (on Minerva 1.64) limits the exponent to 4094, so you can PRINT numbers twice as great as you can enter. The JM rom INPUT routine does not have this restriction. PRINT is slow on extreme values, because it performs repeated floating-point multiplication or division by ten to get the number into range.

The mantissa holds in 31 bits the binary digits of the value. The most significant bit is the sign, using two's-complement arithmetic, with the most negative number coming after the most positive, as is the case with signed 16 bit integers.

The Mantissa of MAXIMUM is \$7FFFFFFF, with all bits set except the sign, while the mantissa of MINIMUM is \$80000000, with only the sign bit set. The tiniest possible number has an exponent of zero and a mantissa of one, giving a value of 1.440917 e-626. It will be ignored if you add it to any number that is not similarly infinitesimal. You get a curious 'overflow' error if you try to INPUT such a low value, on JM and JSL1 alike.

Normally (sic) the most significant bit of the mantissa is shifted all the way to the left, with corresponding adjustments to the exponent, in which case it is said to be 'normalised'. This preserves nine digits of accuracy and means there is just one representation for each value.

If values have different exponents, floating point addition and subtraction start by sliding the lesser value to the right till the exponents match. Then they work out the answer, accurate to 32 bits, with one long integer instruction, and shift the value back to the left if necessary. These shifts may lose some or all significant bits, which is why PRINT 1+1e-7-1 gives 9.965152E-8.

That's good enough, to 32 bit accuracy. I explore the issue of floating point precision in more detail in the extra documentation for DIY Toolkit Volume Z.

The SuperBasic interpreter naively compares floating-point numbers by subtracting one from another and testing the sign of the result. The subroutine COMPARE\_FP, near the end of Listing 1, is much faster. It can compare two floating-point values and indicate the greatest in between seven and eleven machine-code instructions, depending on the values.

For even more speed, at a cost of about 30 bytes, remove the BSR.S and RTS instructions and write the routine out twice, in full, at COMPARE2 and COMPARE3. This makes more difference than you might expect as BSR and RTS are comparatively slow. They must transfer four bytes to or from the A7 stack, and discard the next partly-decoded instruction in the 68008 pipeline.

COMPARE\_FP compares a value in registers D0 and D1 with a value in memory, addressed by offset A2 on A6. The offset is vital when the value is inside the SuperBasic memory area, which could move, taking A6 with it. D0.W contains the exponent, 0-4095, and D1.L holds the 32 bit signed mantissa. The routine uses D2, D4 and D5 internally, sparing D3 as that is the count of values to be scanned.

The routine looks at each part of the value in turn, trying to determine which is the highest. It masks the two mantissae together to compare their signs. If EOR (exclusive OR) leaves a negative value, the signs differ and the result is obvious. Otherwise the exponents are compared to see which is the highest; if they match, the only difference is in the mantissae, which can be compared directly in D1 and D5.

The result is indicated by the carry flag C and zero flag Z. C is set if the registers exceed the value in memory. Z is set if both values are the same. MAXIMUM uses the BLS instruction after the call to keep the same value in the registers if either flag is set. MINIMUM uses BCC, fetching a new lowest value unless carry is clear, indicating that the memory value is less than that in D0 and D1.

## ☆ sinclair QL ☆

### EEC LTD THE WORLD'S LARGEST QL RETAILER

The expandable system for businesses, beginners and experts.

COMPLETE QL's - JS ROM - BRAND NEW PRODUCT - 6 MONTHS GUARANTEE

Package includes PSU, TV Lead, QL Software 2.35: Quill - Wordprocessor, Abacus - Spreadsheet Archive - Database & Easel - Business Graphics, Comprehensive instruction manual.

One year's free membership to QUANTA - the independent user group who provide help, newsletters & a library of 400+ programs - mostly free.

£125

### BARGAIN PRICE MONITORS

All prices include 17.5% VAT

To keep your costs down, the price you see is the price you pay -

WE HAVE BEEN SEARCHING FOR A HIGH QUALITY QL MONITOR FOR A LONG TIME. WE THINK THE GOLDSTAR IS THE ONE!

#### OFFER 1 - colour

GOLDSTAR 14" Hi-Res with 0.31 dot pitch in stylish grey non glare screen (also O.K. for PCs) (Tilt & Swivel base £5) **NEW!** £199

#### OFFER 2 - colour

Philips 8833 Mk II - a lower resolution monitor than the GOLDSTAR but with stereo sound capabilities. 0.42 Dot Pitch. £249

OFFER 3 - Mono-DIGIVISION with QL Connector 12" Green Screen monitor, 2 Colour case **NEW!** £49

### QL TO PC KEYBOARD ENHANCEMENTS

QL/PC 102 keyboards - £25 UK version, - £35

QL/PC keyboard interface - £75 External fitting box & cable - £17

### PRINTERS Centronics - Serial Interface £27 if reqd. All prices include 17.5% VAT

OlivettiTA dot matrix, up to 200cps/30 NLQ, FX850 emulation - £129

Canon BJ10ex bubblejet, very high quality (360 dpi), A4 size - £259

OKI microline, ML183, wide carriage dot matrix printer - £129

We can supply any make/model printer - please ask for details.

QL/Spectrum spares & repairs - please ask for details

### EEC LTD

18-21 MISBOURNE HOUSE,

CHILFERN HILL,

CHALFONT ST PETER,

BUCKS.

SL9 9UE

Tel: 0753 888866

Payment terms:

CWO, Access, VISA et cetera

Cheques - allow 10 days

Delivery:

Carriage - £9 Postage - £3

Product subject to availability. E&OE

Fax: 0753 887149

## C.G.H. SERVICES

Cwm Gwen Hall, Pencader, Dyfed, Cymru, SA39 9HA  
(Tel. 0559 384574) (9am - 5pm)

### PUBLIC DOMAIN AND SHAREWARE LIBRARY

ALL DISKS £2.00 EACH INCLUSIVE OF MEDIA AND P&P

ADVENTURE GAMES DISK 1 (includes Fantasia and Ye Classical Type Adventure)  
ADVENTURE SOLUTIONS DISKS 1-3 (includes The Pawn and Mortville Manor, mainly ST)  
ADVENTURE SOLUTIONS DISK 4 (QL specific)  
ADVENTURE GAMES SOURCE CODE DISK 1 (includes Fantasia, Haunted House, etc)  
ADVENTURE UTILITIES DISK (includes Quill to SuperBasic converter and demo adventure)  
AUSTRALIAN P.D. DISKS 1 + 2 (includes a wide variety of games, utilities etc)  
COMMUNICATIONS DISK 1 (includes QBOX Bulletin Board system)  
COMMUNICATIONS DISK 2 (includes QL Kermi)  
CONNECTIONS DISK 1 (includes ST - QL screen and file converters)  
DEVICE UTILITIES DISK 1 (includes archiving, compacting, formatting and Shell programs)  
DILWYN JONES DISK 1 (includes many SuperBasic progs and Wordsearch)  
EDITORS DISK 1 (includes QED and MicroEmacs)  
EDUCATIONAL PROGRAMS DISK 1 (includes maths, music and chemistry programs)  
EMMANUEL VERBEECK DISK 1 (includes screen save and print progs, and many more utilities)  
ESOTERICA DISK 1 (includes DIY Pyramid construction prog, Biorythms and Psychology progs)  
FRACTALS DISK 1 (includes large numbers of mandelbrot and other fractal progs)  
FRACTALS DISK 2 (Carl Gromin's mandelbrot prog plus animation screens)  
FRACTALS DISK 3 (Rainer Kowalik's mandelbrot prog as amended to give 'Jewel' effect)  
GAMES DISK 1 (includes Starburst, Cavern Frenzy)  
GAMES DISK 2 (includes many arcade type games)  
GAMES DISK 3 (includes QL War)  
GERMAN TO ENGLISH DICTIONARY (multi-tasking)  
GRAPHIX ANIMATION DISKS 1,2,3,4,5 (TV Movies)  
GRAPHIX DEMOS DISK 1,3,4  
GRAPHIX SCREENS (GIF FORMAT) DISKS 1,2,3  
GRAPHIX SCREENS (PIX FORMAT) DISK 1  
GRAPHIX SCREENS (SPECTRUM) DISK 1  
GRAPHIX SCREENS (ST) DISKS 1,2,3,4,5,6,7,8  
GRAPHICS/SCREEN UTILITIES DISK (includes sprite designer, CAD and window progs.)  
HAM RADIO DISK 1 (includes PC conversions)  
INDEXES DISK 1 (Quanta and QL UserWorld)  
MATHS DISK 1 (includes calculator, Pi and calendar creator progs)  
MONEY DISK 1 (Capital Gains Tax and Simple Book Keeper)  
OLIVER FINK DISK 1 (includes progs requiring Pointer environment to work)  
PRINTER UTILITIES DISK 1 (includes label creators and printer tutorials)  
PROGRAMMING DISK 1 (includes QL PROCOD)  
PROGRAMMING DISKS 10-18 (Release 2168 Compiler)  
RALF BIEDERMANN DISKS 1 + 2 (excellent collection of programs, games, utilities, etc)  
RECREATIONAL MATHS DISK (includes life cellular automata etc)  
SUPERBASIC UTILITIES DISK 1 (includes large numbers of utilities progs, toolkits, procedures, etc)  
TEXT DISKS 1-7 (The Bible - Old Testament)  
TEXT DISK 15-16 (The Bible - New Testament)  
TEXT DISK 14 (400+ Business Letters from U.S.A.)  
TYPING TUTOR (Artp - Touch typing tutor)  
THOR DISK 1 (Collection of Thor compatible programs)

Please note the above are only part of our P.D. Library. New programs are constantly being added and more are always welcome. Please send an SAE for full catalogue. We can also supply commercial software for STs and Amigas!  
ALL PRICES INCLUDE MEDIA AND POST AND PACKING FOR THE UK. PLEASE ADD 10% FOR ORDERS IN EUROPE;  
20% OUTSIDE OF EUROPE. PLEASE INDICATE 3.5/5.25" VERSIONS WHEN ORDERING.

# SOFTWARE FILE

## Rich Mellor tests two rare programs for disassembling machine code.

Ever since computers were first invented, they have worked in machine code which in its native form (a series of binary digits) is nigh impossible to understand or program. Before the advent of Basic, assembly language (which uses a series of mnemonics to represent the computer's internal codes) was invented. This was, and still is, a sort of halfway house, where a mnemonic represents a specific instruction that the computer has to carry out, in direct contrast to high-level languages such as SuperBasic.

In a high-level language, you can tell the computer to PRINT something and it will then carry out the hundred or so single instructions it has to perform to take the character you have given it, convert it into the nine lines of pixels which make up the character and chosen ink colour, calculate where the character is to appear on screen, then poke the necessary values onto the screen; whereas it is up to you to do all of the work in machine code.

Although the QL's operating system Qdos has thankfully provided machine code programmers with sets of useful routines, if you have ever tried to program in assembly language you will know how difficult it is to learn.

Arguably, the best way to learn machine code is to take other people's programs and try to adapt them to suit your

```

F1 STATUS F2 DISA 0 Jo 1990
dc.b dc.w dc.l prcl grcl depp text code ascii F3 LOAD F4 SAVE F5 EATOR
DISA - LISTING: pip2_font_ext filetype: 0
Errors: 0
DATA 0
START EQU $
LEA L000A,A1
MOVEA.W $00000110,A2 ; BP_INIT
JMP (A2)
DC.W $0001
DC.W L001C-$
DC.B $06,'S_FONT'.500
DC.L $0000
DC.W $0000
DC.W $0000
BSR.S L0062
BNE.C L0060
BSR.S L003E
SUBA.L A2,A2
  
```

Figure one: an example of DISA at work

needs. However, once you have read articles in magazines and tried to alter machine code, you begin to realise that if you are to take machine code programming seriously, you not only need an assembler to convert assembly language into machine code, but you also need a program which works the other way around, converting the numbers which form machine code into the original assembly language (the source code). This is useful if you ever lose source code, or wish to see how a certain toolkit extension works.

### Assembly

A program which converts machine code into assembly language is called a disassembler. The QL has not been blessed with many dedicated disassemblers, and there are mainly only monitors/disassemblers on the market, such as *Qmon II*. Although this type of program is very good for debugging your own software, it can be a pain if you want to create source code for a program which you do not know in great depth - **Figure Three** shows an example of its output. Once you have the output from a program like *Qmon II*, you will need an editor to strip away the unwanted location numbers (the first column); to remove 'opcodes' (the figures in the second column); to

make sense of lines like the second and third lines; and to convert some of the addresses into labels (where the machine code refers to itself). These two recent releases from the Continent are aimed at making disassembly much easier.

**DISA**, published by Jurgen Falkenberg, is described as a disassembler with an intelligent editor. What this means is that you have total control over the output of the program, but in an interactive way.

The program relies on the QJump pointer interface to create a friendly operating environment. Although the latest versions of the pointer interface and window manager are supplied on the disk, I was rather disappointed that the user is assumed to know how to install this and then call the main program, because no boot program is provided. A small boot program would have been a great help, especially since the pointer interface files have to be loaded in a certain order.

Once and the main program is loaded, the user is presented with the main screen which has the pointer on top of the LOAD command. Pressing space or enter goes to a sub-menu from where you must enter the name of a machine code file to load (no directory is available - but the pointer interface soon lets you do a directory from SuperBasic/Qpac2), or the start address of machine code in

### INFORMATION

**Program:** DISA  
Disassembler (1.0)  
**Supplier:** Jurgen Falkenberg,  
Computer  
Technik, Thanweg  
36,7536 ERSINGEN,  
Germany  
**Memory:** 256K  
**Price:** 69 DM (approx  
£30) + 15 DM p&p  
(3.5" Disk Only)

**Program:** DEA Disassembler  
(v4.07)  
**Supplier:** Ergon Development,  
c/o Davide  
Santachiara, Via  
Emilio de Marchi  
n.2, 42100 Reggio  
Emilia, Italy.  
**Price:** £29 + £5 p&p  
(payable to D  
Santachiara. Add  
£4 for sterling  
cheques;  
nothing for Euro-  
cheques in Lire.

memory. Once the file has been loaded, the program sets up an index of all of the different flags, labels and other internal information which DISA uses to access different parts of the machine code program quickly as it is disassembled.

Once the file has been loaded, the main screen is split into four columns. The first column is empty on start up and its use becomes apparent once you start to disassemble the machine code. The second column contains the offset of each instruction line from the start of the code. 'L' appears in front of some of these offsets to show that they are in fact labels referred to elsewhere in the source code (eg program branches). The next column contains the bytes which make up each instruction (the opcodes), and the final column contains the assembly language instruction itself (if it makes sense).

You must now proceed to scroll down the listing and decide where the machine code



type alters from 'code' to 'data', 'definition block', or 'text'. At the start of each block type, you will need to choose the relevant type and then assign it to the line. At the top of the screen appear the different code types, which you can either select by moving the pointer on top of them and pressing space, or by pressing the letter which is underlined in respect of each type. You then assign the type to a line of machine code by moving the pointer onto the line and pressing Enter. The chosen type appears in the first column against the given line. This does not take as long as you would think, since you only need to specify the lines where the code type alters. Pressing F5 to go to the Editor window will check that you have assigned the correct code type (this window displays the output of the program as it will appear in the final listing).

### Scroll

If the chosen code type is a definition block or text, the program will earmark where the definition block or text ends as a final check that you are using the correct code type. Although the definition block will automatically spot if the block is a Basic (BP\_INIT) list, rom header, Thing header, or Qptr Config block; I was disappointed that the Text code type only copes with text where it is preceded by a word length (byte length text is also quite common).

Because the changes in code type are unlikely to fall at the start of each line on the display, you are able to split lines on the screen on a word boundary. This is much simpler than the manual seems to suggest, in that you merely need to press Space (the line under the pointer will turn green), and then slide the pointer along the opcodes, until a box appears with a little arrow pointing downwards. Pressing Enter while this box is displayed will perform the split (a similar operation is performed to join lines together, although this only works with lines which have previously been split).

Other commands to help you to sort out the machine code program include setting byte, word or longword length data,

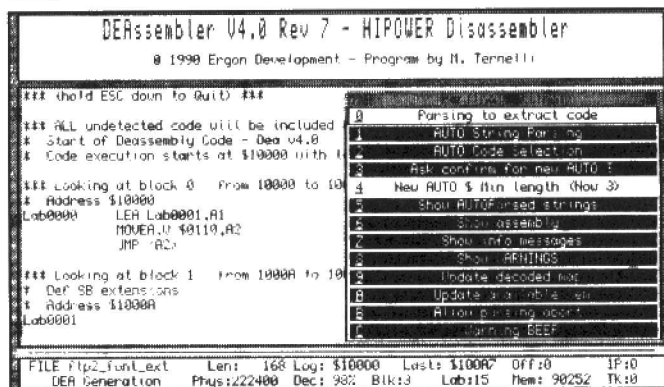


Figure two: DEA working through the code

declaring address or PC relative pointers and setting markers.

I was a little confused by the manual's explanation of markers, until I realised that you can set markers relatively easily by pressing Space on a line in the listing, then moving the pointer onto the Code-Offset and pressing Enter. Once a marker has been set, you can jump to the next marker by switching to the Editor mode (press F5) and press Enter in the middle of the window. These markers are useful for navigating about the listing as it is produced.

screen); and you are unable to move backwards through a listing other than by using the scroll bars.

You can save your work at any time together with the index information to date, but once you have completed working on your listing, you can prepare it for subsequent loading into an assembler.

The F1 (Status) window allows you to dictate remark characters; whether or not addresses are listed together with (PC) after them; whether Qdos or SMS2 trap/vector names are displayed by the program;

when run through my assembler, produced the same machine code as the original file.

Overall, the program is very useful and provides an excellent means of controlling the disassembly of a machine-code program with ease. Unfortunately the documentation leaves something to be desired – a little more information on setting the program up in the first place and using the pointer system would have been a big help. The config program is included to allow you to configure DISA (you can mainly alter the defaults in the F1 Status window), but there is nothing in the documentation concerning its use apart from telling you that it is supplied.

**DEAssembler (DEA)** – published by Ergon Development, is aimed at a slightly different market to DISA, since it will 'automatically' produce a disassembly of the specified machine code. This program will appeal to those less well-versed in machine code since it does not require as great an understanding of the program to be disassembled as DISA.

### Manual

The program is supplied with a sizeable A4 printed manual, which goes into some depth to introduce the program and the idea behind an 'intelligent disassembler'. The manual supplied for review was difficult to understand in places due to the translation from Italian to English, but I have been informed by Ergon that they have asked someone in the UK to improve the manual.

The program is written in Turbocharged SuperBasic and loads quickly from the supplied boot program, presenting the user with an initial menu to load machine code from a file, from memory or from the rom socket. Selecting 'file' allows you to enter a filename, although entering 'flp1\_?' will produce a directory of flp1\_ so that you can remind yourself of the name of the file.

The program is operated by a menu system developed by Ergon themselves, based on the Qlump Pointer system menus, but can be used by a program compiled with Turbo. The

# DISA&DEA

Navigation about the disassembly is a little complicated, because the scroll bar in the main window does not have any effect. Instead you can scroll up or down by a page or by a line at a time, or you can switch to the Editor mode, move to the desired position and then return to the main window (pressing F5 or Space on the Editor command retains your position in the listing). Oddly, in the Editor mode, the scroll bars work fine, but there were two problems with navigation here, in that if you went to the start of the listing (when disassembling a file loaded from disk), the top of the scroll bar went to somewhere slightly above the start of the listing (printing rubbish on the

whether the output is in Capslock or not; and whether or not conditional branches are shown indented in the listing. You are even allowed to insert two assembler directives in the header.

DISA then quickly proceeds to print out the listing to your chosen device. I liked the program's final output, which displays the names of all vectors, Traps and Error codes used by the machine code (as remarks). The assembler directives are useful, but I still found that for the MetaComCo assembler, I needed to delete the DATA directive at the top of the listing (this cannot be altered to a remark line without a separate editor). The output was clear, and thankfully,

menus each have a cursor bar which can be moved up or down onto the desired option, as well as letters against each option, which allow you to select an option by pressing the relevant key. As with Qptr, pressing Space will select an option pointed to by the cursor bar and on most menus will actually carry out the option. However, some menus are specified 'multiple option menus', whereby the desired options must be chosen using the cursor bar or option keys, and then Enter pressed to carry out the function itself.

## Smoothly

The menus operate extremely smoothly and make the program very easy to use for everyone (not just those used to Qptr) – the manual goes into some depth to explain the use of the menus and how to operate the various options.

Once you have specified the machine code to be disassembled, the range of options presented to the user seem mystifying. However, generally the main option required by the user first of all will be 'global parsing'. When selected, DEA will go through the machine code and follow each branch, allocating a label as it progresses. The user can select whether or not to watch DEA as it processes the machine code, as well as selecting the degree of automatic decision-making which DEA can make. This process is quite fast, with DEA automatically spotting definition tables and headers.

The progress of DEA is shown graphically at the bottom of the display in a 'pixel map' which turns green as more of the program is disassembled. Once DEA has followed all of the program's branches and main code, it will then look through the machine code for either byte or word length strings, either automatically selecting them or asking for confirmation by the user as specified.

In stage 3 of the parsing the user has to be involved. This is where DEA attempts to look at any labels which it has allocated but which are not necessarily branches of the machine-code program. Again the

user can decide upon the degree of automation, but will generally have to select the code type relevant to each block of code (you select this from another menu). This process can take time depending on how the machine code program has been written, as well as its length, but DEA places the cursor bar on what it considers the most probable selection for the given block of code. I was a little disappointed that there was not an option for a 'trial' mode here, since it is too easy to identify something as a block of machine code when it is in fact a block of data.

DEA will then continue to parse these code blocks as far as possible, generally decod-

disassembled, you can request DEA to produce the final listing. This is done in a form which can be re-assembled without any alterations – the user can specify the character for Remarks, whether (PC) appears in the listing, and the name used for labels. You can decide whether or not to include code which has not been disassembled (data), or whether to include the Ascii equivalent for such code. The listing itself is then easy to follow, with different code blocks clearly marked (together with the offset from the start of the program).

Overall, the program is easy to use and has many features to examine different parts of a

although it should not be used as a means of breaching copyright.

I liked the way in which the program was controlled, and especially the fact that the user could specify an assumed start address for the machine code, which allowed DEA to disassemble an address dependent program in memory with ease. The labels attached to the disassembled program are then relative to this assumed start address, enabling the user (with a little work) to transform the machine code into address independent code which can be run with LRESPR.

The ability to decide the degree to which DEA can make decisions about the machine code program is something which is essential to this type of program and something which a rival program, *IDIS Special Edition*, sorely lacks. This program is certainly the most versatile machine code programming utility I have seen on any computer to date.

As a test for the relative merits of the two programs, I used a short (168 byte) machine code toolkit extension that I had written some time ago. Both DISA and DEA enabled me to obtain the original source code without much trouble, although DEA did not actually need me to intervene in the disassembly at all.

I preferred the final output from DISA because of the added remarks stating the names of Traps and Vectors used by the program. The indentation for conditional branches together with the tabulated print-out, helped to make the final listing more readable, but this was a little marred by the fact that the label for the branch was not similarly indented.

Both programs provide a useful set of utilities to the machine code programmer. Both programs are very well written, although they suffer from being written in Monitor mode (this affects DEA much more than DISA if you only have a tv). Of course it will be up to the budding machine code programmer to decide which program is most appropriate, although I preferred DEA, mainly because it required a lot less knowledge of the program to be disassembled.

```
0FD382 95CA SUBA.L A2,A2
0FD384 5343 SUBQ.W #1,D3
0FD386 6708 BEQ.S #FD390
0FD388 5343 SUBQ.W #1,D3
0FD38A 6632 BNE.S #FD3BE
0FD38C 2471 MOVE.L #4(A1,A6.L),A2
0FD390 2271 MOVE.L #0(A1,A6.L),A1
0FD394 7025 MOVED #25,D0
0FD396 76FF MOVED #$FF,D3
0FD398 4E43 TRAP #3
0FD39A 4A80 TST.L D0
0FD39C 4E75 RTS
0FD39E 3478 MOVE.W #118.W,A2
0FD3A2 4E92 JBR (A2)
0FD3A4 6616 BNE.S #FD3BC
0FD3A6 4E75 RTS
```

Figure three: A sample printout from QJump's Qmon.

ing around 80% of a large program without any intervention by the user.

If you are disassembling a large machine code program, you may find that the dataspace set aside to store the labels and block information becomes full. However, DEA spots this and will ignore branches to other parts of the program until it has more space available, whereupon DEA will return to those parts which it had to ignore. As with DISA, the disassembly may be saved part-way through processing so that it may be resumed later.

Once the main section of parsing has taken place, DEA dumps the areas which have not been looked at, so that the user can look for the start of any new blocks of code which the program has missed.

Once you are satisfied that the machine code has been fully

machine code program. The options available to the user are various and help to make program disassembly quick and easy, although the defaults which are selected by the program are generally intelligent enough to allow even the least experienced machine code programmer to obtain a source code for simple machine-code programs.

One extra feature of the program is a 'keywords manager', which allows the user to extract certain keywords from a toolkit. Unfortunately this part of the program is not too easy to master, as it is difficult to specify keywords which are already linked into the QL's memory unless you know where they are located. This feature, once mastered, can provide users with a useful means of extracting commands from toolkits for their own use,



# SOFTWARE FILE

## INFORMATION

**Program:** *Remind-me*. A fast and compact date-reminder program.

**Supplier:** Dilwyn Jones  
Computing  
41 Bro Emrys  
Tal-y-Bont  
Bangor  
Gwynedd

**Price:** £12.00. Mdv, 3.5 or 5.25in disks.

Works on an unexpanded QL.

**T**hose of you who have met Joe Haftke will know that he has a thorough and meticulous approach to all the projects he gets involved with.

In developing *Remind-me*, this admirable trait has continued and now we have for the QL one of those 'Diary Manager' programs which are used extensively throughout business and commerce.

### The date

On BOOTing up you are quickly confronted by a screen which asks you to verify the present time/date. In addition, there is a display showing the amount of free memory available. Having verified or corrected the present date, the screen changes to show you all the Events which should be of interest to you at that present time. In addition there is the facility to do a hard copy of the page.

Not a great deal to that, you may say, but *Remind-me* has a few tricks up its sleeve.

When the program was designed, it was decided that not only the current dates should be shown, but also any entry occurring up to seven days prior, and all events and reminders right up to the end of the current month in which you are working.

Therefore, if the program was run on 4th December 1991, it would show all entries from 28th November until 31 December.

SCHEDULE OF EVENTS			DESCRIPTIONS TRUNCATED TO FIT SCREEN		
1. TV LICENCE.	1000/11/30	19. QUANTA AGM.	1991/11/24	37. OUR ANNIVE.	1000/6/10
2. NOT EXPIRE.	1000/14/1	20. ALL FORMAT.	1991/6/22	38. JOHN'S BIR.	1000/7/3
3. PAY CLUB F.	1000/6/10	21. VISIT FRAN.	1992/5/6	39. SUE'S BIRT.	1000/6/5
4. PAY RAC	1000/7/16	22. START HOLI.	1991/7/8	40.	1000/1/1
5. PAY QUANTA.	1000/12/31	23.	1000/1/1	41.	1000/1/1
6. CAR INSURA.	1000/1/1	24.	1000/1/1	42.	1000/1/1
7. PASSPORT E.	1998/10/29	25.	1000/1/1	43.	1000/1/1
8. HOUSE INSU.	1000/11/1	26.	1000/1/1	44.	1000/1/1
9. CONTENT IN.	1000/7/15	27.	1000/1/1	45.	1000/1/1
10. FREEZER IN.	1000/3/20	28.	1000/1/1	46.	1000/1/1
11. HEALTH INS.	1000/12/30	29.	1000/1/1	47.	1000/1/1
12.	1000/7/1	30.	1000/1/1	48.	1000/1/1
13.	1000/12/28	31.	1000/1/1	49.	1000/1/1
14.	1990/12/20	32.	1000/1/1	50. PAY MORTGA.	1000/6/30
15.	1000/1/1	33.	1000/1/1	51.	1000/1/1
16.	1000/1/1	34.	1000/1/1	52.	1000/1/1
17. SEASON TIC.	1000/0/14	35.	1000/1/1	53.	1000/1/1
18. HOUSEKEPT.	1000/0/1	36.	1000/1/1	54.	1000/1/1

ENTER No. OF EVENT TO EDIT (0 TO ABORT)

# REMIND ME

John Shaw sets his computer alarm clock.

*Remind-me* can hold up to 54 different events scheduled by the user which should be enough for most domestic uses. Dates can be scheduled as 'Each Year' or 'Each Month' as required.

The editing of Events is simplicity itself. The whole thing is menu driven and error trapped. Even if you try to enter an event on, say, the 13th month, then the program doesn't crash; you are just politely reminded by a comment on the schedule, such as "ERROR IN DATE".

Joe suggests that the *Remind-me* is best used as a preamble program to your own most frequently used one, for example *Taskmaster*. When you have finished viewing your Diary, the EXIT option continues by BOOTing up the *Taskmaster* boot automatically once the disks have been swapped.

All the usual Customising facilities are available, including default drive, printer port and Baud rate so all users should be catered for. In addition, a dummy diary of events is included so that you can

practice on your backup copy. It also comes with a comprehensive eight page manual which fully explains everything necessary to get the most out of the program. Newcomers to the QL are catered for and instructions on cloning are included.

### Miss no more

*Remind-me* is a very professional, useful and well-thought-out utility. Now there will be no more missed birthdays, insurance renewal dates or Quanta workshops.

# Systematic Machine Code Programming

**In part Alan  
Bridewell writes two  
8049 routines, for  
sound and  
keypresses.**

**T**his series has been building a library of small chunks of assembler language code. Each chunk will do a recognisable job within a program, and be fully annotated, so that it will be clear exactly how to join it to other chunks of code to make a program, and what changes may be necessary to make the chunks fit together.

In this part I am going to look at a couple of bits of programming that may, at first sight, appear to have little in common. These are making (and killing) sounds, and detecting key presses. The connection between these is that the main microprocessor in the QL, the 68008, has nothing directly to do with them. They are dealt with by a separate microprocessor, the 8049. In the QL, it is referred to as the independent peripheral chip (ipc). This means that it can get on with its work of controlling the keyboard, the sound generation, and the microdrives, as well as other, undocumented things, without continually referring back to the 68008. Every now and then it will respond if the 68008 tells it to do something, but otherwise it will just get on with its jobs.

## It own rom

Actually, the 8049 is more like a simple computer in a chip than simply a microprocessor, because it has its own rom of instructions and ram for data built into it. Unfortunately, we cannot access it directly, but have to tell the 68008 to do it for us. This is done by a trap 1 call which simply requires ##\$11 in register D0, and in address register A3 it requires the start address of the commands for the 8049. That's where the real difficulties begin.

For a start, the format of these commands must be exactly correct, or the system will probably crash, if not worse (more about this in a moment). The second complication is that the 68008 and the 8049 can only communicate a nibble four

bits or half a byte at a time, and the command format reflects this fact. The actual command format is as follows. (This next bit is complicated, and you may wish to skip it, unless you really want to understand the details of the listings, rather than simply use them.)

## The first byte

The first byte is the command to be sent to the 8049. (Actually, the command is in the lower nibble of the byte, which means, I assume, that commands are numbered \$0 to \$F.)

The second byte contains the number of bytes of parameters being sent with the command. If this is not correct for the

command, it will almost certainly crash the system.

After these first two bytes comes a long word which describes how the parameters are to be interpreted. This is the most complicated part. Each pair of bits refers to a parameter byte (as there are sixteen pairs of bits in a long word, that must mean a maximum of sixteen parameter bytes). If the pair is 00, it means send the lower nibble of the parameter byte. If it is 10, it means send both nibbles of the parameter byte. If it is either 01 or 11, send nothing of the byte. For example, the sound command requires eight bytes, worth of parameters, and the whole of all the bytes is to be sent. The bit pattern of the long word is 00000000000000001010101010101010

### Listing 1

```

; *****
; 'KEYROW'
; *****
; THIS COMMAND READS A KEYROW, AND REPLIES WITH THE KEY(S) IN THAT ROW
; WHICH ARE PRESSED. IT CAN BE USED TO TEST FOR ANY NUMBER OF
; COMBINATIONS OF KEYS WITHIN THE SAME KEYROW.
; BUT IT WILL NEED TO BE REPEATED TO TEST FOR KEYS IN DIFFERENT KEYROWS.
;
; FIRST WE PUT IN A DELAY. THIS IS TO PREVENT EXCESSIVE USE OF THE IPC.
; THE AMOUNT OF DELAY SHOULD BE ADJUSTED BY TRIAL AND ERROR TO GIVE THE
; BEST RESULT.
; KEYROW          MOVE.W    #$FFFF,D0    ; LOOP 64K TIMES
;                  NOP                      ; DO NOTHING
;                  DBRA      D0,KEYROW      ;
;
; NOW THE KEYROW ROUTINE PROPER
;                  LEA.L     KEYR,A3        ; KEYROW PARAM TABLE IN A3
;                  MOVEQ     #$11,D0        ; $1PCOM IN D0
;                  TRAP      #1              ;
;
; IF THE REPLY IS TO BE TESTED MORE THAN ONCE, MAKE A COPY
;                  MOVE.B    D1,D2          ; COPY REPLY TO D2
;
; ALTER NEXT LINE TO TEST THE REQUIRED ROW(S)
; BY USING DIFFERENT LOGICAL COMPARISONS AND OTHER BRANCH CONDITIONS
; A WIDE VARIETY OF KEY PERMUTATIONS CAN BE TESTED FOR.
;                  CMPI.B    #2,D2          ; IS COLUMN 1 PRESSED?
;                  BEQ.S     ROUTINE1       ; IF SO, BRANCH TO ROUTINE1
;
; ALTER NEXT LINE TO TEST THE REQUIRED ROW(S)
;                  CMPI.B    #4,D1          ; IS COLUMN 2 PRESSED?
;                  BEQ.S     ROUTINE2       ; IF SO, BRANCH TO ROUTINE2
;
; THE LAST FEW LINES CAN BE REPEATED AS MANY TIMES AS YOU NEED TO TEST
; DIFFERENT COMBINATIONS OF KEYS IN THE SAME KEYROW. THE ONLY THING YOU
; MUST REMEMBER IS TO KEEP COPYING THE TRAP CALL REPLY FROM D1 TO D2.
;                  BRA.S     NOKEY          ; ELSE BRANCH TO NOKEY
;
; IF YOU WISH TO KEEP TESTING THE KEYBOARD UNTIL A KEY IN THIS ROW IS
; PRESSED, THEN 'NOKEY' WILL BE 'KEYROW'. BUT IF YOU WISH TO MOVE ON, IT
; WILL BE THE NEXT BIT OF CODE AFTER THE PARAMETER TABLE.
;
; KEYROW PARAMETER TABLE
; KEYR             DC.B      9              ; READ KEYROW
;                  DC.B      1              ; ONE PARAMETER
;                  DC.L      0              ; LOWEST FOUR BITS ONLY
;
; ALTER NEXT PARAMETER TO REQUIRED KEYROW
;                  DC.B      0              ; TEST ROW 0
;                  DC.B      2              ; REPLY OF 8 BITS IN D1
;
; *****

```



010. This converts to the hex number \$AAAA.

Next comes the parameter bytes to be sent. Obviously, the number of bytes here must be the number indicated in the second byte of the command.

Finally, we have a byte which tells the ipc the length of the reply expected. This byte must have a value of 1 for 'no reply expected', or 2 for 'reply of eight bits expected'. I don't know if others are acceptable, but these are the only ones I know to have been documented.

## Sixteen commands

Although there can be up to sixteen commands, numbered \$0 to \$F, the documentation only deals with three: \$9 to read the keyboard, \$A to make a sound, and \$B to kill a sound. In fact, the QL user is sternly warned not to try others for fear of most dire consequences! The chances of finding out what other commands might be used for is remote. In order not to crash the system, you need to put the correct number of parameter bytes in the correct format, and also guess whether a reply is expected. This leaves a large number of random combinations to try before finding out anything useful. However, if the temptation to try a bit of probing in this direction is too great, I strongly recommend that you disconnect anything that might conceivably be damaged before taking a leap into the dark. This would include making sure that no cartridges or floppies are in the drives, and disconnecting your hard disk drive if you have one. At one time, I did try a bit of probing but never succeeded in doing any more than crashing the system, or doing nothing that I could detect. By the way, if any readers do have information about these other commands, I would be interested in hearing from them.

## Our three

The three commands that we can deal with are handled in **listings one, two and three**. The main part of all of them follow exactly the same format. First, the address of the command parameter table is loaded

Listing 2

```

; *****
; 'SOUND'
; *****
; THIS ROUTINE WILL PRODUCE A SOUND USING THE QL SOUND GENERATOR.
; ADJUST THE PARAMETERS IN THE TABLE AS REQUIRED.
; NOTE THAT ALTHOUGH THE PARAMETERS ARE ENTERED AS THOUGH THEY CONSIST OF
; 8 BYTES, THEY ARE ACTUALLY TWO BYTES, FOLLOWED BY TWO WORDS, FOLLOWED BY
; 4 NIBBLES - WHICH COMES TO THE SAME THING.
;
; FIRST WE PUT IN A DELAY. THIS IS TO PREVENT EXCESSIVE USE OF THE IPC.
; THE AMOUNT OF DELAY SHOULD BE ADJUSTED BY TRIAL AND ERROR TO GIVE THE
; BEST RESULT.
.SOUND      MOVE.W    $FFFF,D0 ; LOOP 64K TIMES
.PAUSE      NOP        ; DO NOTHING
            DBRA      D0,PAUSE ;
            LEA.L     SOUNDS,A3 ; SOUNDS PARAM TABLE IN A3
            MOVEQ     #$11,D0 ; #MT_IPCOM IN D0
            TRAP      #1        ;
            BRA.B     NEXTBIT   ; SKIP PARAMETER TABLE

;
; TABLE OF PARAMETERS FOR SOUND
; THE FIRST THREE PARAMETERS MUST NOT BE CHANGED
.SOUNDS      DC.B      $A        ; INITIATE SOUND COMMAND
            DC.B      8          ; 8 PARAMETERS
            DC.L      $AAAA      ; ALL PARAMETERS ARE 8 BITS
; THESE CAN BE ADJUSTED TO THE REQUIRED SOUND
            DC.B      50         ; PITCH 1
            DC.B      100        ; PITCH 2
            DC.W      $2000      ; INTERVAL BETWEEN STEPS
            DC.W      1000       ; DURATION
            DC.B      $11        ; STEP NIBBLE + WRAP NIBBLE
            DC.B      0          ; RANDOM NIBBLE + FUZZ NIBBLE
; THIS MUST NOT BE CHANGED
            DC.B      1          ; NO REPLY
;
; *****

```

into address register A3. Next \$11 is loaded into data register D0 which tells the 68008 that this is a command for the 8049. Then we make the TRAP 1 call. If there is a reply (which there is if we are reading the keyboard), we look for it in D1, and take appropriate action. If there is no reply (and there is none when making or killing sounds) we simply jump to the next bit of code. This much is easy. Most of the care needs to be taken in getting the parameter table right.

Keyrow (**listing one**) concerns command \$9 which reads the keyboard. If you are familiar with the KEYROW command in SuperBasic, you should have no problems with this one. You put in the keyrow of the key or keys you are interested in as a parameter, and the reply is the column value of any keys in that row that have been pressed. For the table of key rows

and columns, look under KEYROW in your *QL User Guide*.

The listing starts with a delay loop. Continually reading the keyboard with this command can cause problems when multi-tasking, because communication with the ipc is rather slow. It is also often unnecessary, and the keyboard can easily be read several times during the course of one fraction of a second keypress, which is clearly a waste of processor time. By adjusting the delay loop by trial and error you can ensure that the keyboard is not read more often than you can produce keypresses!

## Which way

Next comes the trap call which is the same in all these listings, followed by some lines to look at the reply telling us which keys in the keyrow have been pressed, and taking appropriate action. At this point, there is a great variety of ways the chunk of code can go, many of which have identical outcomes. (A full analysis of this would require a whole article on digital logic and Boolean algebra, followed by another on all the conditional branch instructions.) I have put in what seems to me to be the code most likely to be useful. What it does is to assume we will want to test more than one specific combination of key presses in the same row. (For example, in row 7 we might want to test for ALT-X, CTRL-X and CTRL-SHIFT-X, with each leading to different results.) What it does not do is allow the testing for, say, any one of a group of

Listing 3

```

; *****
; 'KILLSOUND'
; *****
; THIS ROUTINE WILL SWITCH OFF THE SOUND FROM THE SOUND GENERATOR.
;
.KILLS      LEA.L     KSOUND,A3 ; KILLSOUND PARAM TABLE IN A3
            MOVEQ     #$11,D0 ; #MT_IPCOM IN D0
            TRAP      #1        ;
            BRA.B     NEXTBIT   ; SKIP OVER PARAMETER TABLE

;
; KILLSOUND PARAMETER TABLE - DO NOT ALTER ANY PARAMETER
.KSOUND      DC.B      $B        ; KILLSOUND
            DC.B      0          ; NO PARAMETERS
            DC.B      0          ; NO REPLY
;
; *****

```

## MACHINE CODE

keys in the same row. (For example, we might want to see if any of the function keys in row 0 has been pressed, without at this stage identifying which one.) This kind of thing can be done by careful use of `ANDI.B` commands instead of `CMPI.B` commands, together with `BNE.S` instead of `BEQ.S`.

## More comparisons

Firstly, if more than one comparison is needed (that is, we're looking for more than one combination of key presses in the same row), we copy the reply to register D2. This can obviously be left out if only one comparison is needed, and must be repeated if we are making more than two comparisons. Next we compare the copy with the column value for the keys we are looking for using the CMPI.B instruction. If the comparison shows we have the combination of keys we were looking for, the BEQ.S command will branch to the appropriate code. If not, the next two lines will test the reply left in D1 for the other combination we are looking for (assuming we are only looking for two at the moment) and branch to the appropriate code if the comparison shows them equal. If none of the combinations we are looking for is found, we then branch to appropriate code with a BRA.S command.

Finally, we have the parameter table, which will be as follows:

byte	\$9 - read the keyrow
byte	\$1 - one parameter byte
longword	\$0 - read lowest four bits only
byte	- parameter (number of
keyrow	to be tested)
byte	\$2 - reply of 8 bits expected in
D1	

Obviously, the only part of this that you will want to alter is the parameter byte, which you adjust to the keyrow you require.

## Sound loop

Sound (**listing two**) concerns \$A which makes a sound. As with Keyrow, this starts with a delay loop, and for much the same reason. There is no point in trying to start a new sound before the last one has had a chance to get going! The parameters here are quite complicated, and need some care. There are actually eight bytes worth of sound parameters, but they are divided into two bytes (pitch 1 and pitch 2), followed by two words (step interval and duration), followed by four nibbles (step, wrap, randomness and fuzziness). Assemblers cannot handle separate nibbles of data, so step and wrap are combined into a byte of data, as are randomness and fuzziness. So the complete parameter table will be as follows:

#### Testing 4

```

*****
JOBSTART
      BRAS      START      ; BRANCH TO START OF CODE
      DC.L      0          ; THIS IS STANDARD FORMAT FOR
      DC.W      $4AFB      ; START OF A JOB)
----- ALTER CHARACTER COUNT AND JOB NAME -----
      DC.W      11         ; CHARACTER COUNT OF JOB NAME
      DC.B      'CURSOR$OUND' ; NAME OF JOB

*****
PRIORITY
----- ALTER LABEL TO .START -----
.START      MOVEQ      #$B,D0      ; #MT_PRIOR IN D0
            MOVEQ      #-1,D1      ; OF THIS JOB
            MOVEQ      #1,D2       ; TO 1
            TRAP       #1          ;

*****
KEYROW
      KEYROW     MOVE.W      #$FFFF,D0      ;
      .PAUSE     NOP
            DBRA      D0,PAUSE
            LEA.L      KEYR,A3      ; KEYROW PARAM TABLE IN A3
            MOVEQ      #$11,D0      ; #IPCOM IN D0
            TRAP       #1
; ---- NEXT 3 LINES TO COPY REPLY AND TEST NOT NEEDED, SO DELETE ----
; ---- ALTER NEXT LINE TO TEST REQUIRED COLUMNS ----
            CMPL.B      #14,D1      ; ARE COLUMNS 1, 2 AND 3 PRESSED?
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO KEYROW2 ----
            BEQ.S      KEYROW2      ; IF SO, MOVE ON
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO KEYROW ----
            BRAS      KEYROW      ; ELSE READ AGAIN

; KEYROW PARAMETER TABLE
      .KEYR      DC.B      9          ; READ KEYROW
            DC.B      1          ; ONE PARAMETER
            DC.L      0          ; LOWEST FOUR BITS ONLY
; ---- ALTER NEXT PARAMETER TO TEST ROW 7 ----
            DC.B      7          ; TEST ROW 7
            DC.B      2          ; REPLY OF 8 BITS IN D1

*****
KEYROW
; ---- ALTER LABEL IN NEXT LINE TO KEYROW2 ----
; ---- ALTER NEXT LINE TO GIVE A SHORTER DELAY ----
      .KEYROW2     MOVE.W      #$8000,D0
; ---- ALTER LABEL IN NEXT LINE TO PAUSE2 ----
      .PAUSE2      NOP
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO PAUSE2 ----
            DBRA      D0,PAUSE2
; ---- ALTER LOAD ADDRESS IN NEXT LINE TO KEYR2 ----
            LEA.L      KEYR,A3      ; KEYROW PARAM TABLE IN A3
            MOVEQ      #$11,D0      ; #IPCOM IN D0
            TRAP       #1
            MOVE.B      D1,D2      ; COPY REPLY TO D2
; ---- ALTER NEXT LINE TO TEST REQUIRED COLUMN ----
            CMPL.B      #4,D2      ; IS COLUMN 2 PRESSED?
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO HISOUND ----
            BEQ.S      HISOUND
; ---- ADD NEXT 6 LINES TO TEST OTHER COLUMNS ----
            MOVE.B      D1,D2
            CMPL.B      #12B,D2      ; IS COLUMN 7 PRESSED?
            BEQ.S      LOSOUND      ;
            MOVE.B      D1,D2      ;
            CMPL.B      #6B,D2      ; ARE COLUMNS 0 AND 6 PRESSED?
            BEQ.S      KEYROW      ;
            CMPL.B      #9,D1      ; ARE COLUMNS 0 AND 3 PRESSED?
            BEQ.S      QUIT      ;
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO KEYROW2 ----
            BRAS      KEYROW2      ;

; KEYROW PARAMETER TABLE
; ---- ALTER LABEL IN NEXT LINE TO .KEYR2 ----
      .KEYR2      DC.B      9          ; READ KEYROW
            DC.B      1          ; ONE PARAMETER
            DC.L      0          ; LOWEST FOUR BITS ONLY
; ALTER NEXT PARAMETER TO TEST KEYROW 1
            DC.B      1          ; TEST ROW 1
            DC.B      2          ; REPLY OF 8 BITS IN D1

*****
BOUND
; ---- ALTER LABEL IN NEXT LINE TO LOSOUND ----
      .LOSOUND     MOVE.W      #$FFFF,D0      ;
; ---- ALTER LABEL IN NEXT LINE TO .PAUSE3 ----
      .PAUSE3      NOP
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO PAUSE3 ----
            DBRA      D0,PAUSE
; ---- ALTER LOAD ADDRESS TO LOSOUND6 ----

```



```

LEA.L    LOGSOUNDS,A3 ; SOUNDS PARAM TABLE IN A3
MOVEQ    #11,DO ; #MT_IPCOM IN DO
TRAP     #1
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO KEYROW2 ----
BRA.B    KEYROW2 ; SKIP PARAMETER TABLE

;
; TABLE OF PARAMETERS FOR SOUND
; ---- ALTER LABEL IN NEXT LINE TO LOGSOUND ----
LOGSOUNDS DC.B    $A ; INITIATE SOUND COMMAND
          DC.B    8 ; 8 PARAMETERS
          DC.L    $AAAA ; ALL PARAMETERS ARE 8 BITS
; ---- ALTER NEXT 6 PARAMETERS TO GIVE REQUIRED SOUND ----
          DC.B    255 ; PITCH 1
          DC.B    255 ; PITCH 2
          DC.W    0 ; INTERVAL BETWEEN STEPS
          DC.W    1 ; DURATION
          DC.B    0 ; STEP NIBBLE + WRAP NIBBLE
          DC.B    0 ; RANDOM NIBBLE + FUZZ NIBBLE
; THIS MUST NOT BE CHANGED
          DC.B    1 ; NO REPLY

;
; *****
; 'SOUND'
;
; ---- ALTER LABEL IN NEXT LINE TO HISOUND ----
HISOUND MOVE.W    #FFFF,DO ;
; ---- ALTER LABEL IN NEXT LINE TO PAUSE4 ----
PAUSE4 NOP
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO PAUSE4 ----
DBRA     DO,PAUSE4
; ---- ALTER LOAD ADDRESS IN NEXT LINE TO HISOUNDS ----
LEA.L    HISOUNDS,A3 ; SOUNDS PARAM TABLE IN A3
MOVEQ    #11,DO ; #MT_IPCOM IN DO
TRAP     #1
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO KEYROW2 ----
BRA.B    KEYROW2 ; SKIP PARAMETER TABLE

;
; TABLE OF PARAMETERS FOR SOUND
; ---- ALTER LABEL IN NEXT LINE TO HISOUNDS ----
HISOUNDS DC.B    $A ; INITIATE SOUND COMMAND
          DC.B    8 ; 8 PARAMETERS
          DC.L    $AAAA ; ALL PARAMETERS ARE 8 BITS
; ---- ALTER NEXT 6 PARAMETERS TO GIVE REQUIRED SOUND ----
          DC.B    1 ; PITCH 1
          DC.B    1 ; PITCH 2
          DC.W    0 ; INTERVAL BETWEEN STEPS
          DC.W    1 ; DURATION
          DC.B    0 ; STEP NIBBLE + WRAP NIBBLE
          DC.B    0 ; RANDOM NIBBLE + FUZZ NIBBLE
; THIS MUST NOT BE CHANGED
          DC.B    1 ; NO REPLY

;
; *****
; 'SOUND'
;
; ---- ALTER LABEL IN NEXT LINE TO QUIT ----
QUIT MOVE.W    #FFFF,DO ;
; ---- ALTER LABEL IN NEXT LINE TO PAUSES ----
PAUSES NOP
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO PAUSES ----
DBRA     DO,PAUSES
LEA.L    SOUNDS,A3 ; SOUNDS PARAM TABLE IN A3
MOVEQ    #11,DO ; #MT_IPCOM IN DO
TRAP     #1
; ---- ALTER BRANCH ADDRESS IN NEXT LINE TO END_JOB ----
BRA.B    END_JOB ; SKIP PARAMETER TABLE

;
; TABLE OF PARAMETERS FOR SOUND
SOUNDS DC.B    $A ; INITIATE SOUND COMMAND
        DC.B    8 ; 8 PARAMETERS
        DC.L    $AAAA ; ALL PARAMETERS ARE 8 BITS
; ---- ALTER NEXT 6 PARAMETERS TO GIVE REQUIRED SOUND ----
        DC.B    50 ; PITCH 1
        DC.B    100 ; PITCH 2
        DC.W    $2000 ; INTERVAL BETWEEN STEPS
        DC.W    1000 ; DURATION
        DC.B    $11 ; STEP NIBBLE + WRAP NIBBLE
        DC.B    0 ; RANDOM NIBBLE + FUZZ NIBBLE
; THIS MUST NOT BE CHANGED
        DC.B    1 ; NO REPLY

;
; *****
; 'ENDJOB'
;
JOB_END MOVE.W    $CA,A2 ; UT_ERR0 in A2
        JSR      (A2) ;
;
END_JOB MOVEQ    #5,DO ; #MT_FRJOB IN DO
        MOVEQ    #-1,D1 ; ID OF THIS JOB IN D1
        TRAP     #1
;
; *****

```

byte \$A - make a sound command  
 byte \$8 - 8 parameter bytes  
 longword \$AAAA - Send all 8 bits of all 8 bytes  
 byte - pitch 1  
 byte - pitch 2  
 word - step interval  
 word - duration  
 byte - step nibble + wrap nibble  
 byte - randomness nibble + fuzziness nibble

Clearly the first three parameters must not be changed, but the rest you can adjust to the sound you require. This is not the place to go into the ins and outs of QL sound generation, but suffice to say it is somewhat peculiar. However, if you can produce the sound you want using SuperBasic, you should have no trouble converting the parameters in listing two to the ones you require.

## No delay here

Killsound (listing three) simply does that - kills any sound the QL may be making. There is no need for a delay loop here, because this code is unlikely to be repeated in rapid succession except by mistake! As the command carries no parameters and no reply is expected, the parameter table consists only of the following (and obviously needs no alteration):

byte \$B - command to kill a sound  
 byte 0 - no parameters  
 byte 0 - no reply

Now to apply this new-found knowledge. Clearly the combination of detecting key presses and making sounds leads one to think of designing programs to make the QL behave like an electric organ. Although this is not particularly difficult, it must be pointed out that sound production is not one of the QL's strong points, and the results are likely to be disappointing. It is probably more fruitful to use the QL sounds to inform you about key presses. For instance, it would not be difficult to write a routine that made a little sound each time a key was pressed.

## Multitasking

This could then multi-task with another program, say *Quill*, and it would reassure you that the keyboard was, in fact, still active, even though Quill was going through one of its 'I must sit back and think about this one' phases when you ask it to do something complicated with a large document. However, a word of warning about this. The ipc communication is rather slow, and continually scanning the whole keyboard can slow things down considerably.

A better option might be to simply scan certain important keys. For instance, if you

are like me, you may be staring at the screen trying to scroll down through a document, and finding it doesn't work, because you are at the top of the document pressing the UP key. Here is a routine to get over this. Quite simply, EXEC it before EXECing Quill or Abacus, and it will give a short high note when the UP key is pressed, and a short low note when the DOWN key is pressed. However, the program can do rather more than this.

One other very useful application of the keyrow routine is that it enables you to produce the Hotkey effect. You can EXEC a program with a low priority, so it sits in the background doing nothing but checking for a particular combination of keys to be pressed. As soon as it detects this, it jumps into life, and starts doing its job. (It can even increase its priority to do it if necessary.) Then when it has finished what it was doing, it can go back to sleep (by lowering its priority again) and simply wait for same key press combination to occur again. We shall incorporate this into our program.

## Statements

As before, rather than write a flow chart, I shall write a series of statements about the program, together with the names of the chunks of code, and any important points to be made.

1. First, we start in the usual way with JOBSTART, with an appropriate job name.

2. Because we want our program to multitask with word processors and spreadsheets without slowing them down too much, we use PRIORITY to set the priority to 1.

3. Next we use KEYROW to test for a particular key combination to wake up the program. I have used CTRL-ALT-X, because it is an obscure combination, unlikely to be needed in the other multitasking program, and also the three keys are in the same row so one KEYROW call will do. This will simply loop round if the keys are not pressed, and because it is a short loop, I have put the maximum delay time in it.

## Key combination

4. Once our key combination has been detected, the main program is activated, and we move to a second KEYROW. This is altered to test for four specific key combinations in row 1 and take appropriate action. If the up arrow is pressed, it branches to give a short high sound. If the down arrow is pressed it branches to produce a short low sound. If SPACE ENTER is pressed, it branches back to the first KEYROW, which effectively puts the program to sleep again. If ESC ENTER is pressed, it branches to a 'quit' routine.

5. Next we use SOUND three times, and adjust to give appropriate sounds. The first is the up arrow routine with a short high note, and the second is the down arrow routine with a short low note. Both these must finish by branching back to the second KEYROW ready for another key press. The third is the start of the 'quit' routine, so the program makes a distinctive sound when it has been told to kill itself.

## In at the kill!

6. Finally we end with ENDJOB to kill the program. This is not strictly necessary, but it is bad form to have a program which cannot kill itself, unless it is to be called from another job which can kill it.

So, putting all that together, we get:  
JOBSTART, PRIORITY, KEYROW,  
KEYROW, SOUND, SOUND, SOUND,  
ENDJOB.

---

*"It is bad form to have a program which cannot kill itself unless it is to be called from another job which can kill it."*

---

Listing four shows the result of merging the required chunks of code into a file and making the appropriate alterations to them to make them fit together. As before, most of the comments have been removed, with lines of asterisks left between the chunks to show clearly where one ends and the next begins. Also, additional comments have been added to point out the necessary alterations. You could, of course, simply copy listing four, and assemble it, but it would defeat the point of the exercise. It is much more informative to start with the chunks of code, merge them together, and then make the alterations, and note why those alterations are made.

Having got the program to work, you could then experiment with it. Try altering the sounds – you might find the ones you prefer. You could also test for other keys, and give them suitable sounds. If you are a bit more adventurous, you could get the program to display a small window in an unwanted part of the screen (if you can find one!) to tell you the program is loaded, and whether or not it is active.

In part 2 of this series, we put together a

program to transfer bytes of data from one channel to another, for instance, to print a file, or to copy a file. This program is an excellent candidate for the Hotkey treatment, so it can be brought to life with the right key combination, and then put to sleep again when it has finished. The modifications needed are as follows.

1. After Jobstart and Priority, the program goes into a loop looking for a particular key combination. (This should preferably be an obscure, non-printing one.) ALT-X would be good, because ALT and X are in the same row, which make it easier.

2. On detecting the required key combination, it jumps into the main routine. You could increase the priority if you wanted, but it is probably no advantage in this program.

3. On completing the file transfer, it closes the channels as before. But now, instead of going to Endjob, it jumps back to the loop to detect the key press combination to start it up again.

4. If this is all you did, it would never get to Endjob to kill itself, so Endjob could be deleted from the program. In this case, the program would remain active in the background until you killed it with a toolkit command, or reset the machine. This is rather crude programming, as all multitasking programs should be removable from within the program when it is finished with. There are a couple of reasonable ways to do this in this case. One is to print a message asking if the program should be aborted when the file transfer is completed, and instructing the user to press ALT-Y to kill the program or ALT-N to just put it to sleep. It could then detect the appropriate key presses for either a jump to Endjob, or a jump back to the start. The other way is to use the loop at the start to check for a key combination (ALT-ESC, for example) to jump to Endjob as well as check for ALT-X to jump into action. Both alternatives have good and bad points, and it is really a matter of taste.

## Two more things

Two other things you could do to the transfer program to make use of the sound generating potential. You could (a) get the program to make a suitable rude noise instead of an error message if Open cannot open the channel, and (b) get the program to make a more pleasant sound on completion of the file transfer.

So there it is. Although we have only increased our little 'Assembler Toolkit' by three routines, and only used two of them in our examples, I hope you can see that they are very useful ones. With good use, they can improve the user friendliness of your program by quantum leaps. (Haven't I heard that somewhere before, many years ago...)

Happy coding!



# DJC

## Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,  
Bangor, Gwynedd LL57 3YT U.K.  
Tel: Bangor (0248) 354023

### GRAPHICS

**THE PAINTER V4.04** ..... £25.00  
[F 512K] 100% machine code art and graphics program, by PROGS of Belgium. Multiple screens, can be mouse controlled, selection of fonts, patterns, many facilities.

**THE CLIPART** ..... £12.00  
[F 128K] 3 disks full of clipart pictures, for use with most DTP/graphics programs.

**QRACTAL** ..... £20.00  
[F 512K] Machine code fractals program. Displays Julia or Mandelbrot sets.

**QRACTAL SCREENS** ..... £ 5.00  
[F 128K] A disk full of sample screen pictures generated with the QRactal program. Buy this to see the sample pictures, then ask about a discount when buying QRactal itself!

**IMAGE PROCESSOR V2** ..... £15.00  
[F 256K] Image enhancement, edge detection, zoom and edit, touch up picture generated by digitisers, scanners or other graphics programs. Works with mode 4 or 8 screens.

**UPGRADE OLD IMAGE PROCESSOR** ..... £10.00  
(Send proof of purchase of old version)

**PD2 CLIPART** ..... £10.00  
[F 128K] Clipart made for PD2 Plus, but available in screen format for use with other DTP/graphics programs if required.

**SCREEN SNATCHER** ..... £12.00  
[F 1M 128K] 'Grab' screen displays from within other programs that are not able to save their own displays.

**TEXT 'N' GRAPHIX** ..... £20.00  
[F 256K] **NEW!** Allows you to insert screens, or part of QL screen pictures, into a Quill or plain text file printout. Mix NLQ text and graphics in the same printout! Currently only for use with 9 pin Epson compatible dot matrix printers - ask about a version for 24 pin printers due soon.

**TRANS24** ..... £10.00  
[F 1M 128K] Translates 9 pin graphics printouts into data for a 24 pin printer.

### DTP

**PAGE DESIGNER 2 PLUS** ..... £40.00  
[F 512K] QL Desktop Publishing program, mix text and graphics, make printed pages or posters, use clipart, use graphics from other programs etc. Prints on most printers. Ask for details!

**UPGRADE PD2 TO PD2 PLUS** ..... £20.00  
(Send proof of purchase of old PD2)

### DATA BASES

**ADDRESS BOOK & LABEL PRINTER** ..... £15.00  
[F 2M R 384K] Store names and addresses and print them out on a variety of label sizes, or print a telephone list, etc.

**DATA DESIGN** ..... £50.00  
[F 512K] Superb, fast 100% machine code database by PROGS of Belgium. Programmable via BASIC and machine code interfaces - write your own database applications in BASIC using the DATA Design commands to help you! Runs in pointer environment (supplied)

**FLASHBACK** ..... £25.00  
[F 1M 256K] Fast machine code database which is also very easy to use.

**FLASHBACK SPECIAL EDITION** ..... £40.00  
[F 2M 512K] Enhanced version of Flashback, new commands, report generator, etc

**UPGRADE TO FLASHBACK S.E.** ..... £20.00  
(return Flashback master disk)

**QL GENEALOGIST, STANDARD VERSION** ..... £19.50  
[F 2M 384K] Family trees and family history database, one of our best selling programs.

**QL GENEALOGIST SECOND EDITION** ..... £30.00  
[F 384K] Enhanced version - for details see our advert in QL World December 1991.

**UPGRADE TO SECOND EDITION** ..... £12.00  
(return Genealogist master disk)

**BUDGET 128K QL GENEALOGIST** ..... £12.00  
[F 1M 128K] Cut down version for unexpanded machines.

**SUPER DISK INDEXER** ..... £12.00  
[F 1M 384K] Catalogue your disk collection as a database of files. Store it, print it, search & find files, make your life easier.

### QL HARDWARE

**MINI PROCESS CONTROLLER** ..... £59.95  
Relay switched outputs, controlled via QQL serial port.

**SOFTWARE TOOLKIT FOR MPC ABOVE** ..... £9.95  
[F 128K] Control the MPC from BASIC programs more easily with these extensions.

**NETWORK PROVER** ..... £ 3.50  
Plugs between network lead and computer to give a visual indication of transmission.

POSTAGE RATES - SEE BELOW

### TEXT

**BIBLE TEXT DISKS, EDITOR FORMAT** ..... £20.00  
**BIBLE TEXT DISKS, "DOC"** ..... £20.00  
[F 256K] Text of the King James Bible on disk. Please state whether you require Editor (plain text) or Quill "\_doc" format

**SPELLBOUND** ..... £30.00  
[F 1M 384K] A spelling checker which can check spelling **as you type!** 30,000 word dictionary.

**SPELLBOUND SPECIAL EDITION** ..... £50.00  
[F 512K] Enhanced version of Spellbound with 50,000 word dictionary & many new features.  
**UPGRADE TO SPELLBOUND S.E.** ..... £30.00

**QUICK POSTERS**  
[F 2M 128K] Text poster maker, for use with Star NL, XB and LC printers ..... £10.00

**ROB ROY BARGAIN PACK** ..... £10.00  
[F 3M 128K] Reviewed in QL World August 1991

### OTHER QL SOFTWARE

**HOME BUDGET** ..... £20.00  
[F 1M 128K] Domestic bills and accounts program, plus a UK Income Tax Calculator.

**REMIND-ME** ..... £12.00  
[F 1M 128K] Dates and events reminder program - remember about birthdays, licence renewals, etc. Quick and very easy to use.

**SCREEN ECONOMISER** ..... £10.00  
[F 1M 128K] Turns off the QL display after a set number of minutes to protect the screen.

**SLOWGOLD** ..... £ 5.00  
[F 128K] Slowdown routine and control panel for software which runs too fast on Gold Card or indeed any QL system.

**TASKMASTER** ..... £25.00  
[F 1M 384K] Task switching utility. Enables you to conveniently switch between several programs in memory. Calculator, notepad and file handling utility included.

**SUPER DISK LABELLER** ..... £10.00  
[F 256K] Print neat labels for your floppy disks listing the filenames on the disk in columns in small print.

**THE CAT** ..... £ 5.00  
[F 1M 128K] List files on a disk or cartridge to the screen, or print on paper, in columns (optionally sorted in this latest version). Useful and convenient utility.

★ SEE ALSO THE OTHER HALF OF THIS ADVERT ON THE PRECEDING PAGE ★



SOFTWARE POSTAGE: Software sent post-free to UK addresses. Abroad add £1.00 per program for postage and packing.  
SUPPLIES POSTAGE: For disks, boxes and stands, add £2.50 for postage to UK addresses, or 10% of order value (minimum postage £3.50) for airmail postage where possible.  
PAYMENT: We can accept payment by cash, cheque (in UK Pounds Sterling only, drawn on UK branch of bank or building society), Postal Order, International Postal/Money order, or Eurocheque. We can also accept payment by these credit cards: VISA, ACCESS, MASTERCARD, EUROCARD and also Barclays CONNECT card. Please state your card type, number, expiry date and your address (goods paid for by credit card can only be sent to cardholder's address). Remember to sign your order. Goods remain property of DJC until paid for in full. Our telephone number above has an answering machine for when we are unable to answer in person - don't be shy!

